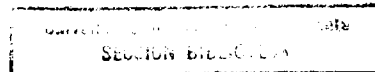


CURSO DE CONTROL DIFUSO

FRANCISCO JAVIER ALVAREZ ECHEVERRY

GILBER CORRALES RUBIANO



031772



C.U.A.O.
BIBLIOTECA



0039388

T0001232

**CORPORACION UNIVERSITARIA AUTONOMA DE OCCIDENTE
DIVISION DE INGENIERIAS
PROGRAMA DE INGENIERIA ELECTRONICA
SANTIAGO DE CALI
1999**

CURSO DE CONTROL DIFUSO

FRANCISCO JAVIER ALVAREZ ECHEVERRY

GILBER CORRALES RUBIANO

**Monografía para optar el título de
Ingeniero Electrónico.**

Director

Doctor

FREDDY NARANJO PEREZ

Ingeniero Mecánico

CORPORACION UNIVERSITARIA AUTONOMA DE OCCIDENTE

DIVISION DE INGENIERIAS

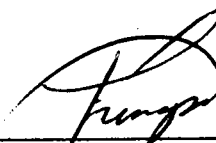
PROGRAMA DE INGENIERIA ELECTRONICA

SANTIAGO DE CALI

1999

Donación de Francisco Javier Alvarez 66-70-21 9 2210018

Nota de aceptación



Presidente del Jurado

Oscar Mauricio Agudelo.
Jurado

J. F. Ali
Jurado

Santiago de Cali, Abril de 1999

621.39
9473c
1.1

CONTENIDO

| | Pág. |
|---|-------------|
| INTRODUCCIÓN | 1 |
| 1. INTRODUCCIÓN LOS SISTEMAS DE CONTROL | 3 |
| 1.1.EL PROBLEMA DE CONTROL | 3 |
| 1.2.SOLUCIONES CONVENCIONALES AL PROBLEMA DE CONTROL | 4 |
| 1.2.1. Control clásico | 5 |
| 1.2.2. Control moderno | 12 |
| 1.3.CONTROL INTELIGENTE | 15 |
| 1.3.1. Antecedentes | 15 |
| 1.3.2. Definición y aspectos básicos | 16 |
| 1.3.3. Control inteligente vs. Control convencional | 20 |
| 1.3.4. Técnicas de control inteligente | 22 |
| 1.3.4.1.Sistemas expertos | 22 |
| 1.3.4.2.Redes neuronales | 24 |
| 1.3.4.3.Algoritmos genéticos | 27 |
| 1.3.4.4. Lógica difusa | 29 |
| 2. INTRODUCCION A LA LÓGICA DIFUSA | 31 |
| 2.1. DESARROLLO HISTÓRICO | 31 |
| 2.2. CONJUNTOS DIFUSOS | 36 |
| 2.3. OPERACIONES ENTRE CONJUNTOS DIFUSOS | 39 |
| 3. SISTEMAS DIFUSOS | 43 |
| 3.1. INTRODUCCIÓN | 43 |
| 3.2. VARIABLE LINGÜÍSTICA | 49 |
| 3.3. REGLAS IF-THEN DIFUSAS | 51 |
| 3.4. FUZZIFICACIÓN | 54 |
| 3.5. BASE DE REGLAS Y MOTOR DE INFERENCIAS | 55 |

| | |
|--|------------|
| 3.6. DEFUZZIFICACIÓN | 59 |
| 3.7. APLICACIONES | 60 |
| 4. CONTROL DIFUSO | 62 |
| 4.1. DESARROLLO HISTÓRICO | 63 |
| 4.2. ESTRUCTURA BÁSICA DE UN CONTROLADOR DIFUSO | 66 |
| 4.2.1. Interface de fuzzificación | 67 |
| 4.2.2. Base de conocimiento | 69 |
| 4.2.3. Motor de inferencia | 71 |
| 4.2.4. Interface de defuzzificación | 73 |
| 4.3. CONTROL DIFUSO DIRECTO | 77 |
| 4.4. CONTROL PID DIFUSO | 77 |
| 4.5. CONTROL DIFUSO POR MODOS DESLIZANTES | 80 |
| 4.6. CONTROL DIFUSO ADAPTATIVO | 84 |
| 4.7. OTROS ENFOQUES | 89 |
| 5. ESTABILIDAD Y ROBUSTEZ DE CONTROLADORES DIFUSOS | 94 |
| 6. APLICACIONES DEL CONTROL DIFUSO: CASOS | 99 |
| 6.1 APLICACIÓN DE UN SISTEMA DE CONTROL HÍBRIDO PID-DIFUSO PARA UNA PLANTA PETROQUÍMICA | 99 |
| 6.2 CONTROL DE UNA PLANTA INCINERADORA DE BASURAS MEDIANTE TÉCNICAS DIFUSAS | 103 |
| 6.3 LIMITADOR DE VELOCIDAD PARA CAMIONES DE CARGA | 106 |
| 7. HERRAMIENTAS DE SOFTWARE PARA DISEÑO, SIMULACIÓN E IMPLEMENTACIÓN DE CONTROLADORES DIFUSOS | 111 |
| 8. DISPOSITIVOS DE HARDWARE PARA LA IMPLEMENTACIÓN DE CONTROLADORES DIFUSOS. | 163 |
| 9. CONCLUSIONES | 174 |
| BIBLIOGRAFIA | 176 |
| ANEXOS | 183 |

LISTA DE TABLAS

| | Pág. |
|---|-------------|
| Tabla 1-1 Arreglo de Routh | 6 |
| Tabla 1-2 Especificaciones de control clásico | 11 |
| Tabla 3-1 Aplicaciones de lógica difusa en control | 61 |
| Tabla 4-1 Desarrollos Históricos de Control Difuso | 64 |
| Tabla 4-2 Arreglo FAM | 71 |
| Tabla 4-3 Aplicaciones de Control Difuso | 76 |
| Tabla 7-1 Herramientas de Software | 111 |
| Tabla 7-2 Clasificación de Software | 162 |

LISTA DE FIGURAS

| | Pág. |
|--|-------------|
| Figura 1-1 Diagramas de Nyquist | 7 |
| Figura 1-2 Lugar geométrico de las raíces para un sistema de segundo orden | 8 |
| Figura 1-3 Lugar geométrico de las raíces | 9 |
| Figura 1-4 Gráficos de Bode | 10 |
| Figura 1-5 Diagrama Estructural de un Sistema Experto | 23 |
| Figura 1-6 Estructura de una neurona | 25 |
| Figura 1-7 Red neuronal artificial | 25 |
| Figura 2-1 Relación entre plataformas | 35 |
| Figura 2-2 Función de pertenencia del conjunto difuso A | 39 |
| Figura 2-3 Representación gráfica de la unión de A y B | 40 |
| Figura 2-4 Representación gráfica de la intersección de A y B | 41 |
| Figura 2-5 Representación gráfica del complemento (\bar{A}) de A | 42 |
| Figura 2-6 Representación gráfica de $A \subset B$ | 42 |
| Figura 3-1 Configuración básica de un sistema de lógica difusa puro | 44 |
| Figura 3-2 Configuración de un sistema Takagi y Sugeno's | 47 |
| Figura 3-3. Configuración básica de sistema de lógica difusa con fuzzificación y defuzzificación | 48 |
| Figura 3-4 Representación de la variable lingüística velocidad | 50 |
| Figura 4-1 Estructura de un Controlador difuso CD | 66 |
| Figura 4-2 Aplicación de un Controlador difuso CD al control de un proceso | 67 |
| Figura 4-3 Proceso de fuzzificación para la variable velocidad | 68 |
| Figura 4-4 Controlador difuso directo (CDD) | 77 |
| Figura 4-5 Controlador PID difuso | 78 |
| Figura 4-6 Controlador PI difuso | 80 |
| Figura 4-7 Trayectoria del estado e vs de . | 84 |

| | |
|---|-----|
| Figura 4-8 Esquema de un controlador adaptable | 85 |
| Figura 4-9 Adaptación difusa de un controlador PI | 86 |
| Figura 4-10 SOC de Procyk y Mamdani | 87 |
| Figura 4-11 Estructura básica de un controlador difuso adaptativo | 87 |
| Figura 4-12 Arquitectura híbrida | 90 |
| Figura 4-13 Controlador Híbrido (intervención del CD sobre el PID) | 91 |
| Figura 4-14 PID precompensado por un Controlador Difuso | 92 |
| Figura 4-15 Modificación difusa de ganancias en PID | 92 |
| Figura 6-1 Proceso de producción | 100 |
| Figura 6-2 Configuración del control híbrido fuzzy-PID | 101 |
| Figura 6-3 Operación con el control híbrido PID-Difuso | 103 |
| Figura 6-4 Estructura de la planta incineradora de basuras | 104 |
| Figura 6-5 Área de observación cubierta por la cámara infrarroja | 105 |
| Figura 6-6 Estructura del sistema de control difuso | 106 |
| Figura 6-7 Curva de velocidad vs tiempo | 107 |
| Figura 6-8 Diagrama del sistema | 108 |
| Figura 6-9 Unidad electrónica de control | 108 |
| Figura 6-10 Estructura del sistema de lógica difusa | 109 |
| Figura 6-11 Variables lingüísticas y funciones de pertenencia | 110 |
| Figura 6-12 Curva de respuesta | 110 |
| Figura 7-1 Desarrollo de un controlador difuso (FLC) | 112 |
| Figura 7-2 Demo del Péndulo Invertido | 123 |
| Figura 7-3 Pantallas de animación y edición <i>FuzzyTech</i> | 127 |
| Figura 7-4 Editor de <i>FuzzyTech</i> | 128 |
| Figura 7-5 Editores GUI's | 144 |
| Figura 7-6 Modelo de un caso general | 145 |
| Figura 7-7 Modelo de un caso específico | 145 |
| Figura 7-8 Uso de editores GUI's | 146 |
| Figura 7-9 Editor FIS | 147 |
| Figura 7-10 Editor GUI de funciones de pertenencia | 149 |
| Figura 7-11 Editor GUI de Reglas | 152 |

| | |
|---|-----|
| Figura 7-12 Bloque de Control difuso para Simulink | 160 |
| Figura 7-13 Simulación del control de nivel de un tanque Bloque (Simulink) | 161 |
| Figura 8-1 Aplicación del microcontrolador en un lazo control | 164 |
| Figura 8-2 Procesador difuso VY86C570 | 165 |
| Figura 8-3 PLC difuso | 168 |
| Figura 8-4 Arquitectura interna NLX 220. | 170 |
| Figura 8-5 Aplicación del microcontrolador 8XC196 en un lazo control | 172 |

LISTA DE ANEXOS

| | Pág. |
|--|-------------|
| Anexo A, Practicas de simulación | 183 |
| Anexo B, Practicas de laboratorio | 198 |
| Anexo C, Ejercicios y problemas | 204 |
| Anexo D, Presentación en diapositivas “Curso de Control Difuso” | 213 |

RESUMEN

El presente trabajo de grado pretende llenar el vacío existente en la formación que reciben los estudiantes de la CUAO y de la región en un área de mucha importancia y de gran actualidad como es el Control Difuso, para tal efecto se realizó una amplia investigación bibliográfica local y nacional, mediante visitas a distintas bibliotecas, e internacional a través de los servicios de la Internet. Toda esta información recopilada fue metodológicamente organizada y sintetizada por capítulos en forma de monografía y complementada con una serie de anexos los cuales terminan por darle el cuerpo completo al curso de control difuso.

INTRODUCCIÓN

Esta monografía recopila a través de ocho capítulos los aspectos más relevantes que deben ser tenidos en cuenta para el desarrollo de un curso de control difuso.

La estructura metodológica de la monografía pretende llevar al lector de los elementos básicos de control convencional pasando luego a comprender la lógica difusa y la teoría de conjunto difusos, para luego conociendo los diferentes tipos de sistemas difusos se pueda comprender la aplicación de la lógica difusa al control. Finalmente se dan las herramientas para el desarrollo de estrategias de control difuso como lo son el software y el hardware.

De esta manera en el primer capítulo se empieza describiendo muy brevemente algunas de las técnicas más comunes de control convencional, haciendo la distinción entre control clásico y control moderno y posteriormente se analiza el concepto de control inteligente. En el segundo capítulo se estudian los conceptos más importantes de la lógica difusa, empezando con su desarrollo histórico y la teoría de los conjuntos difusos propuesta por Zadeh en 1965. En el tercer capítulo se definen los tres tipos de sistemas difusos más importantes, al igual que los elementos de un sistema difuso. Al final del capítulo se hace una referencia muy breve de algunas aplicaciones de sistemas difusos. El cuarto capítulo se constituye en el capítulo central de la monografía ya que abarca el concepto de control

difuso, empezando por su desarrollo histórico y definiendo los conceptos para desarrollar una adecuada estrategia de control lingüística, en él también se tratan brevemente algunas combinaciones de control difuso con otras técnicas de control convencional. En el quinto capítulo se trata la estabilidad y la robustez de los controladores difusos de manera muy breve ya que aún es un tema en investigación. En el sexto capítulo se tratan tres casos de aplicaciones industriales de control difuso, con el fin de dar a conocer implementaciones reales de control difuso. El séptimo capítulo trata las herramientas de software usadas para implementar y simular sistemas de control difuso y finalmente el capítulo ocho recorre las herramientas de Hardware para el desarrollo de controladores difusos de tiempo real.

La monografía incluye cuatro anexos que cubren la implementación de prácticas de simulación y laboratorio, ejercicios por capítulo y una presentación en diapositivas para la exposición magistral del curso.

Se recomienda al lector leer los capítulos en orden para una mejor comprensión. Claro que si se tiene buen conocimiento del control convencional se puede omitir el primer capítulo.

1. INTRODUCCIÓN A LOS SISTEMAS DE CONTROL

El conocimiento de la teoría de control proporciona una base para entender el comportamiento de los sistemas automáticos. Los sistemas convencionales de control emplean un modelo matemático que describe la dinámica del sistema pero debido a la complejidad de las plantas actuales han surgido procesos a los cuales es muy difícil o muy costoso hallarles el modelo matemático que mejor aproxime su dinámica. Para tales sistemas han surgido estrategias de control diferentes a las convencionales, dando paso al control inteligente.

En este capítulo se describen muy brevemente algunas de las técnicas de control más comunes de control convencional, haciendo la distinción entre control clásico y control moderno. Posteriormente se analiza el concepto de control inteligente y se explican las principales técnicas de control inteligente con que se cuenta actualmente.

1.1. EL PROBLEMA DE CONTROL

El problema de control surge cuando al disponerse de un sistema (planta, proceso, máquina, etc.) queremos hacer que su comportamiento, que está reflejado en las variables que lo

caracterizan físicamente, obedezca a nuestras intenciones, lo cual tiene necesariamente que coincidir con lo más conveniente a los fines con que ha sido creado.

1.2 SOLUCIONES CONVENCIONALES AL PROBLEMA DE CONTROL

Para poder realizar el análisis y el diseño de un sistema de control dando solución a un problema planteado, es necesario definir el problema en un lenguaje matemático¹. Una vez sea deducido el modelo matemático del sistema es posible el diseño del controlador mediante alguna de las técnicas convencionales que ofrece la teoría de control. Debe quedar claro que las matemáticas son una **herramienta** mediante la cual se pueden describir, analizar y diseñar sistemas de control. Obtener un modelo matemático más o menos exacto de un sistema es uno de los problemas más importantes en la ingeniería de control.

El primer modelo matemático para describir el desempeño de una planta para propósitos de control se atribuye a James Cleark Maxwell, quien en 1868 usó ecuaciones diferenciales para explicar problemas de inestabilidad encontrados en el regulador de velocidad de Watt. Desde ese entonces se han hecho significativos avances sobre la teoría de control. Con el uso de los métodos en el dominio de la frecuencia y la transformada de Laplace en los años 30's y 40's y con los desarrollos en control óptimo y el análisis de espacios de estados en los 50s y 60's y posteriormente los avances en control estocástico, robusto y adaptativo y con el advenimiento de los microprocesadores, se ha hecho posible desde entonces poder llegar a

¹ MASTER Michal K., ASTROM Kari J., en Modern Control Systems

controlar más eficientemente sistemas mucho más complejos y en mejor forma que el regulador de velocidad de Watt.

Cuando Maxwell usó el modelamiento matemático y mostró que los problemas de inestabilidad del regulador de velocidad de Watt dependían de que las raíces de su ecuación característica tuviera partes reales negativas, demostró la importancia y la utilidad de los modelos matemáticos para el entendimiento de fenómenos complejos, de esta manera señaló el comienzo de la teoría de control.

Los métodos de control convencional se pueden dividir hoy día en ***control clásico y control moderno***.

1.2.1 Control clásico la teoría de control clásico utiliza extensamente el concepto de la función de transferencia, el análisis y diseño se realizan en el dominio de la variable de Laplace (S) y/o en el dominio de la frecuencia. El control clásico brinda muy buenos resultados en los sistemas de una entrada y una salida (SISO, por sus siglas en inglés).

Las herramientas más comunes para el análisis de sistemas mediante estas técnicas incluyen: diagramas de Bode, pruebas de estabilidad de Nyquist y Routh, lugar de las raíces, así como también márgenes de ganancia y de fase, entre otras.

Las técnicas más comunes para los controladores son compensadores en cascada tales como el famoso PID o compensadores en adelante y/o atraso.

Las especificaciones para el diseño de un controlador clásico incluyen estabilidad, un amortiguamiento razonable, rapidez de respuesta y mínimo error en estado estacionario.

A continuación se describirán brevemente algunas de las técnicas más utilizadas para el análisis de sistemas de control clásico:

Criterio de estabilidad de Routh el criterio de estabilidad de Routh es un método en el cual los coeficientes de la ecuación característica en lazo cerrado son organizados y ubicados en una tabla (ver ejemplo 1-1), ésta es examinada y la estabilidad es verificada observando el signo de los parámetros dentro de la tabla.

Ejemplo 1-1, sea la ecuación características

$$S^6 + 3S^4 + 7S^3 + 20S^2 + 6S + 15 = 0$$

El arreglo de Routh sería:

Tabla 1-1 Arreglo de Routh

| Coeficientes | | |
|--------------|----|----|
| 1 | 7 | 6 |
| 3 | 20 | 15 |
| 1/3 | 1 | |
| 11 | 15 | |
| 6/11 | | |
| 15 | | |

Este criterio sólo decide si el sistema es inestable o no, por lo cual su utilidad en el análisis de sistemas es limitada, ya que no decide nada acerca de la estabilidad relativa o sobre cómo estabilizar un sistema inestable.

Existen otras técnicas que no sólo determinan la estabilidad sino que también dan una medida de la estabilidad relativa, es decir sobre qué tan cerca está el sistema de convertirse en inestable, estas técnicas son gráficas, ellas son:

- ▣ Criterio de Nyquist
- ▣ Diagramas del Lugar de las Raíces
- ▣ Diagramas de Bode

El criterio de Nyquist enfoca su atención en la función de transferencia en lazo abierto del sistema, la gráfica que utiliza este método es bidimensional, con el eje horizontal para las partes reales de la función de transferencia y el eje vertical para la parte imaginaria. La variable de Laplace (S) es transformada en su equivalente frecuencial ($j\omega$).

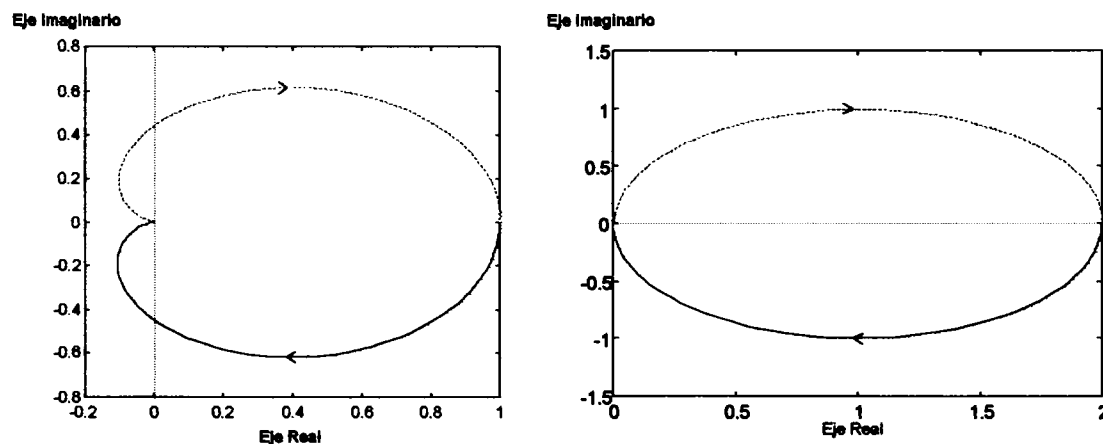


Figura 1-1 Diagramas de Nyquist

El criterio de Nyquist dice “si la característica frecuencial de la función de transferencia en lazo abierto de un sistema lineal e invariable en el tiempo, no envuelve al punto $(-1, j0)$ del diagrama polar, ese sistema será estable, o sea que en lazo cerrado, todos los polos de su función de transferencia tendrán parte real negativa”. En la figura 1-1 se pueden ver algunos

diagramas típicos del diagrama de Nyquist.

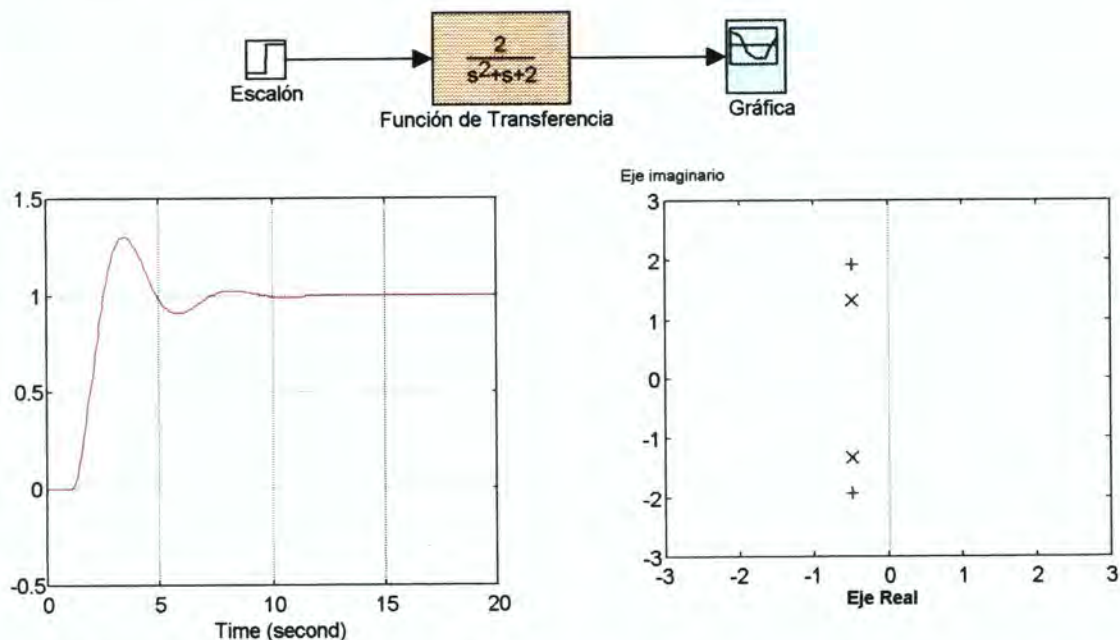


Figura 1-2 Lugar geométrico de las raíces para un sistema de segundo orden.

El diagrama del lugar de las raíces muestra la localización de los polos del sistema en lazo cerrado como una función de la ganancia de la función de transferencia en lazo abierto. Al diseñar un sistema de control lineal este método resulta muy útil, pues indica la forma en que hay que modificar la posición de los polos y ceros de lazo cerrado para que la respuesta cumpla con las especificaciones de comportamiento esperadas del sistema.

Ejemplo 1.2, sean los sistemas de segundo orden mostrados en la parte superior de las figuras 1-2 y 1-3; su correspondiente diagrama del lugar geométrico de las raíces será el mostrado por las gráficas en la parte inferior de cada figura.

Analizando el diagrama del primer sistema podemos ver que tienen dos polos complejos conjugados con partes reales negativas. Concluimos que la respuesta en el tiempo está

caracterizada por un sobre-pico y una oscilación amortiguada, la cual eventualmente alcanza el estado estacionario como se ve en ambas figuras.

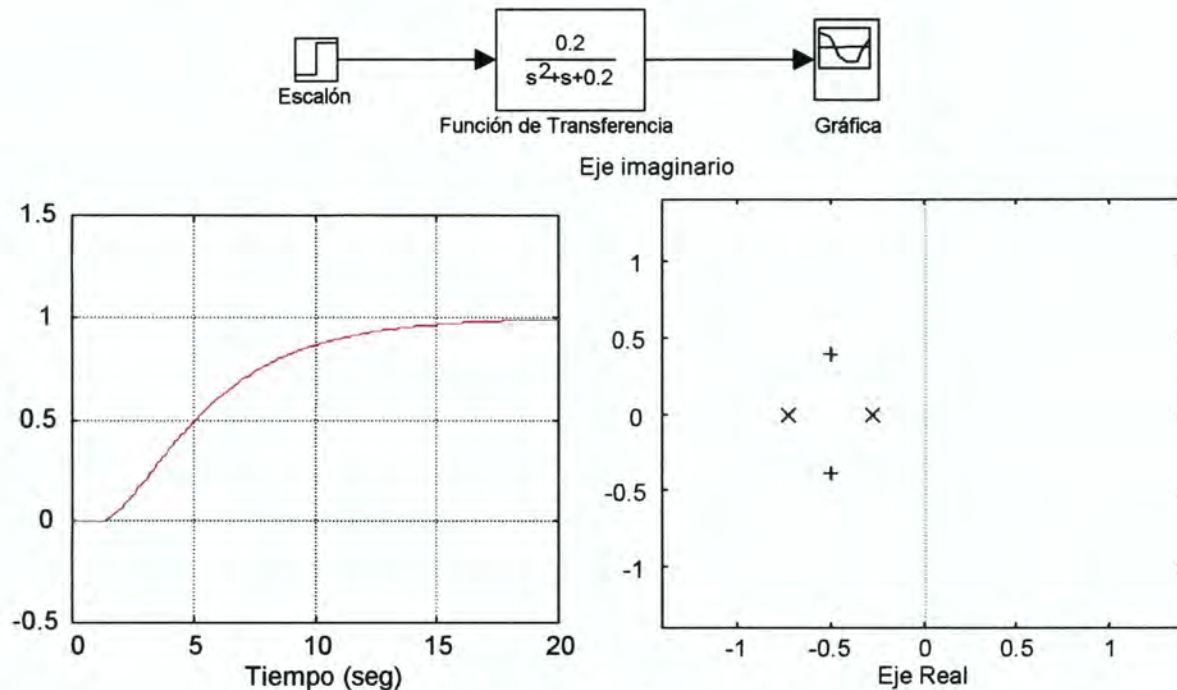


Figura 1-3 Lugar geométrico de las raíces

En la figura 1-3 se observan dos polos reales negativos distintos. El sistema es sobreamortiguado y estable, el comportamiento está caracterizado por un decaimiento exponencial de la constante de tiempo dominante definida por el polo más cercano al origen.

Los Gráficos de Bode son equivalentes a los de Nyquist ya que ambos métodos son representaciones gráficas de la función de transferencia en lazo abierto. Sin embargo, mediante la técnica de Bode el análisis se realiza en dos gráficas separadas, una para la magnitud de la función de transferencia y otra para la fase. Estas dos características son en

función de la frecuencia de operación. Veamos un ejemplo de Bode para la función de transferencia:

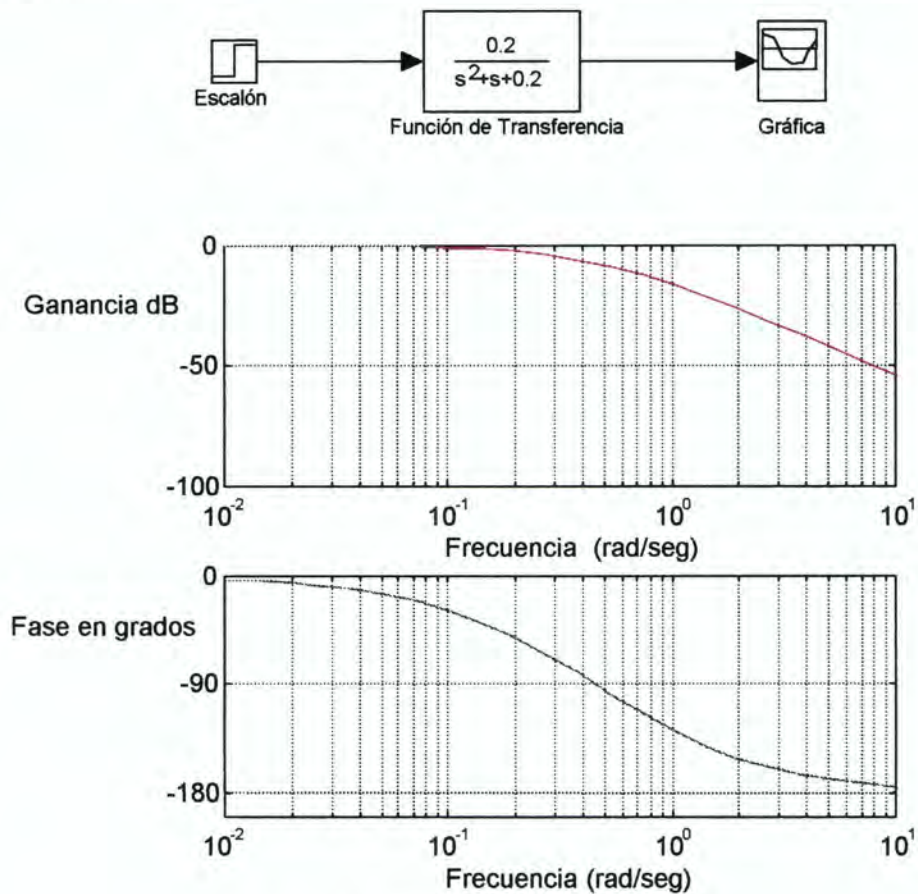


Figura 1-4 Gráficos de Bode

Las anteriores son las técnicas clásicas más utilizadas para el análisis de sistemas de control. Cuando se está diseñando, el objetivo es desarrollar un controlador que lleve al sistema a cumplir con las especificaciones de desempeño requeridas. En control clásico, tales especificaciones están definidas de varias maneras, usualmente se dividen en especificaciones en el dominio de la frecuencia y en el dominio del tiempo. Estas son:

Tabla No 1-2 Especificaciones de control clásico

| Dominios | Especificaciones |
|--------------------------|---|
| Dominio de la Frecuencia | Ancho de banda Margen de ganancia Margen de fase |
| Dominio del Tiempo | Sobre paso Retardo de tiempo Tiempo de rizado Tiempo de establecimiento Constante de tiempo dominante |

En general, el diseño de sistemas de control mediante las técnicas clásicas es un proceso de “ensayo y error”, usando las técnicas de análisis discutidas anteriormente para evaluar los controladores candidatos. Si las características reveladas no reúnen las especificaciones requeridas, entonces el controlador es modificado y el análisis es repetido para determinar si quedan satisfechas.

El método más común para la implementación en el sistema del controlador, es ubicar en cascada un compensador cuya función de transferencia se adiciona a la del sistema afectando sus características.

Los compensadores más comúnmente usados son:

- ❑ Compensador de factor de ganancia
- ❑ Compensador en adelanto

- ☐ Compensador en atraso
- ☐ Compensador en adelanto-atraso o atraso-adelanto
- ☐ PID y todas sus combinaciones

1.2.2 Control Moderno a diferencia del control clásico, el control moderno se basa en técnicas en el dominio del tiempo y permite resolver de la misma manera sistemas SISO y MIMO (de múltiples entradas y múltiples salidas) y, a diferencia de las técnicas clásicas para análisis (Nyquist, Bode, lugar de las raíces, etc.) las técnicas de control moderno sirven tanto para análisis como para diseño.

El diseño de sistemas de control mediante técnicas clásicas involucra un proceso de ensayo y error, los métodos de control moderno comienzan generando una descripción matemática formal del sistema y de los objetivos de diseño, entonces se aplican las técnicas para resolver dicho problema matemático y por consiguiente deducir el controlador que mejor satisfaga las metas del diseño.

Como se dijo antes, el control moderno está caracterizado porque el análisis y el diseño están enfocados hacia el dominio del tiempo, sin embargo, existen algunas técnicas que enfatizan su atención hacia el dominio de la frecuencia (control robusto).

La representación matemática de los sistemas, procesos y/o plantas se hace mediante las *ecuaciones de estado*, las cuales son una representación alternativa de las ecuaciones diferenciales. Las ecuaciones de estado contienen la misma información dinámica, pero

pueden ser generalizadas y por consiguiente ser usadas para generar el diseño de un sistema de control, con una amplia aplicación hacia un variado número de plantas. En control moderno las ecuaciones de estado son expresadas en forma matricial, representando sin dificultad plantas sencillas y complejas. Una planta compleja requerirá matrices de mayor orden. Entre las características de la técnica de espacio de estados que la hacen muy atractiva para resolver muchos sistemas que no podrían analizarse por los métodos clásicos se tienen que:

- ▣ Los sistemas lineales con parámetros variantes en el tiempo, pueden ser analizados de la misma manera que los sistemas lineales invariantes en el tiempo.
- ▣ Sistemas de gran orden pueden ser analizados por las mismas tecnologías que los de bajo orden. Igual sucede con los sistemas MIMO y SISO.
- ▣ La formulación matemática en el espacio de estados es fácilmente programable en un computador para su análisis o para la implementación de un controlador digital.
- ▣ A pesar de que los detalles sean diferentes, la técnica del espacio de estados es un fundamento válido para sistemas no lineales o para sistemas estocásticos.

A continuación examinaremos algunos conceptos fundamentales para el análisis y diseño mediante las técnicas de control moderno.

El Estado de un sistema es el mínimo conjunto de variables que contienen la suficiente información sobre el pasado del sistema y que permiten calcular todos los estados futuros que éste pudiera asumir.



Variable de estado es una función del tiempo cuyo valor en cualquier instante representa el estado del objeto o planta en dicho instante.

Controlabilidad cuando se trabaja con sistemas de control, es necesario hacerse la pregunta: ¿es el sistema controlable?. Es decir, de que si existe una señal(es) de control que permitan que el desempeño del sistema deseado sea llevado a cabo.

En términos más formales se diría que el sistema,

$$\frac{dx}{dt} = Ax + Bu$$

Es totalmente controlable si y solo si la matriz Q de $n \times nm$ tiene rango n , donde

$$Q = [B: AB: A^2B: \dots A^{(n-1)}B]$$

Donde la matriz A describe la dinámica del estado (x) y B describe la relación de la señal de control (u) hacia los estados.

Observabilidad para fines prácticos es útil conocer si el estado del sistema es medible o no para realizar alguna realimentación de éste y poder tomar una acción de control, esta es la característica de *observabilidad* la cual se puede definir que el siguiente sistema,

$$\frac{dx}{dt} = Ax + Bu$$

$$y = Cx + Du$$

Es totalmente observable, si y sólo si la matriz P ($kn \times n$) tiene rango n , donde

$$P = [C : CA: CA^2: \dots CA^{(n-1)}]^T$$

En resumen la prueba de Controlabilidad determina en qué tanto o si es posible controlar una planta en su totalidad y la observabilidad determina si es posible medir un estado usando la salida de la planta.

Entre las ramas de control lineal moderno más difundidas y populares se encuentran: el control robusto, control adaptativo, control óptimo, en ellos se usan técnicas de diseño tales como los filtros Kalman, realimentación del estado, LQR, H_∞ , LQG, etc.

1.3 CONTROL INTELIGENTE

1.3.1 Antecedentes la literatura científica resalta tres proyectos que han formado las bases para los sistemas actuales de control inteligente (CI).

En el siglo 17, Gottfried Leibniz propuso construir una máquina capaz de realizar cualquier proceso de razonamiento por el que se le preguntara.

En el siglo 19, Charles Babbage, con su Máquina Analítica podía realizar operaciones de álgebra y de esta forma pensó en manejar el sistema formal unitario en el cual sólo la pluralidad de los fenómenos físicos puede ser adecuadamente descritos. En 1936, Alan Turing propuso la construcción de una máquina que pudiera pensar. Hoy día, las técnicas de Control Inteligente han sido capaces de llevar a la realidad los sueños de estos tres grandes visionarios.

En las primeras décadas del siglo 20 se hicieron importantes estudios acerca del sistema nervioso, incluido el cerebro, como un elemento de control. En 1930 este trabajo fue finalizado y se dio comienzo a la Cibernética. En los años 50's, científicos como McCarthy, Minsky, Newell y Simón integraron la cibernética con los computadores y hallaron formas para habilitar a la máquina para realizar cosas tales como resolver problemas de geometría, jugar ajedrez, reconocer patrones y voces. Fue así como nació la Inteligencia Artificial (IA).

El Dr. K. S. Fu introduce por primera vez el término “control inteligente” e inicia estudios en esta área. Los esfuerzos más recientes en el desarrollo de sistemas de CI se han concentrado en la integración de la IA con los avanzados algoritmos de control.

1.3.2 Definición y Aspectos Básicos el término Control Inteligente o Sistema de Control Inteligente ha sido descrito indistintamente, pero un aspecto de común entendimiento es que la Inteligencia describe el **grado de autonomía** del sistema de control.

“Inteligencia” es una palabra asociada con la habilidad humana y podría no ser apropiada para caracterizar un sistema de control. De esta forma, en un sentido restrictivo, los sistemas de control comunes tienen que recorrer un largo camino antes de calificar para este nombre². Sin embargo, una interpretación más apropiada para los Sistemas de Control Inteligente se da en términos de Control Autónomo. Pudiéndose definir lo siguiente³:

☐ **Inteligencia** : el grado de autonomía

² MASTER., ASTROM., Op. Cit., p. 4

³ RAHMAT Shoureshi: Intelligents control systems: Are they for real?.

- ☐ Autonomía : auto-independencia
- ☐ Sistema de Control Inteligente: un sistema de control con un alto grado de autonomía en términos de auto-aprendizaje, auto-reconfigurabilidad, razonamiento, planeación y producción de decisiones y con la habilidad para extraer la información más valiosa de datos ruidosos e imprecisos de cualquier sistema dinámicamente complejo y/o sus alrededores.

Veamos el nivel de inteligencia de los siguientes sistemas de control:

1. Controlador Clásico: tiene un bajo nivel de inteligencia, ya que posee una estructura fija, no es auto-reconfigurable, no soporta grandes cambios en los parámetros del sistema y solamente toma su forma de decisión en términos del calculo de la señal de control mediante el error de su referencia.
2. Controlador PID adaptable: tiene un nivel de inteligencia medio ya que posee algún grado para re-sintonizar sus parámetros, pero con una estructura de control fija (PID), tiene alguna capacidad de aprendizaje dentro de la forma de identificación del sistema.
3. Controlador Robusto: también posee un bajo nivel de inteligencia, debido a que tiene una estructura fija, no es auto-reconfigurable y su habilidad para manejar cambios importantes en sus parámetros es debida a la inteligencia del diseñador y no al controlador.

Por otro lado, encontramos otro aspecto para tener en cuenta para la definición de un

controlador inteligente y es desde el punto de vista de la metodología⁴, se dice que existe una Metodología de Control Inteligente cuando se usan técnicas y procedimientos motivados en el comportamiento de sistemas humanos, animales y/o biológicos para diseñar, construir y/o implementar un controlador o una estrategia de control.

De esta manera, podemos concluir que otra definición para un controlador inteligente puede ser:

El dispositivo físico llamado controlador será un controlador inteligente si está desarrollado o es implementado con (a) una metodología de control inteligente o (b) sistemas convencionales o técnicas de control que emulan o desarrollan funciones de control que son normalmente desempeñadas por humanos, animales o sistemas biológicos.

Otra definición para lo que es un sistema de control inteligente es⁵: *"Un sistema es inteligente si tiene la habilidad de actuar apropiadamente en un ambiente incierto, donde una acción apropiada es aquella que aumenta la probabilidad de éxito y donde éxito es el logro de sub-metas de comportamiento que llevan al logro de la meta última del sistema."*

Para Antsaklis *"Inteligencia es una propiedad del sistema que emerge cuando los procesos de enfoque de atención, búsqueda combinatoria y generalización se aplican a la entrada para producir una salida"*. También explica que *"la inteligencia en una máquina es el proceso de analizar, organizar y convertir datos en conocimiento, donde el conocimiento*

⁴ PASSINO Kevin: Toward Bridging the perceived gap between....

⁵ From conventional control to Autonomous Intelligent Methods. En IEEE control systems. Oct-1996

está definido como información estructurada adquirida, usada para remover ignorancia o incertidumbre acerca de una tarea específica pertinente a la máquina inteligente."

Examinemos ahora algunos de los aspectos más generales que deben satisfacer los controladores inteligentes:

Adaptación y Aprendizaje: la capacidad de adaptación es necesaria en un sistema inteligente, pero si se requiere una "mayor" inteligencia es esencial la capacidad de aprendizaje.

Autonomía el sistema de control inteligente debe tener la capacidad de "razonar" acerca del estado actual del sistema y de detectar cambios dentro de su estado interno y los alrededores. Debe ser capaz también de reconfigurar su operación a fin de compensar las fallas que se presenten.

Arquitectura funcional: la complejidad creciente de los sistemas actuales, hace necesaria una estructura apropiada para analizar y evaluar estrategias de control. Esta estructura debe brindar mecanismos de abstracción o al menos facilitar el ordenamiento para reducir la complejidad.

Otro aspecto clave en el estudio de los sistemas de control inteligente es realmente en qué circunstancias o bajo qué situaciones se debe aplicar control inteligente.

Ambientes variables e inciertos: las incertidumbres que pueden manejar las metodologías

adaptables y robustas suelen estar limitadas (relativamente previsibles y estructuradas). Entre mayor sea la incertidumbre respecto a los cambios mayor debe de ser la inteligencia del sistema.

Estructuras de control jerárquico y/o distribuido: las condiciones actuales de la producción flexible, han popularizado este tipo de estructuras; en los procesos puntuales se utilizan generalmente controladores convencionales, pero la planeación y manejo de tiempos y recursos responden a estrategias complejas que habitualmente se desarrollan con altos niveles de inteligencia.

Sistemas Autónomos: la autonomía ha llegado a ser uno de los aspectos vitales en el diseño y desarrollo de sistemas tales como naves espaciales, sistemas de exploración subacuáticas, etc.

Interacción con sistemas biológicos: cada vez más los sistemas automáticos abandonan las fábricas y entran a interactuar con sistemas biológicos que presentan comportamientos poco predecibles, dependientes de aspectos de difícil evaluación, altamente no-lineales y sobre todo muy vulnerables.

1.3.3 Control Inteligente vs. Control Convencional las técnicas utilizadas mediante el control convencional están fundamentadas en el formalismo de las descripciones matemáticas, en donde el nivel del poder de abstracción es bastante grande.

Los métodos de control y las teorías matemáticas fueron desarrollados para satisfacer las

siempre crecientes necesidades de la tecnología. Por esto, cada vez se han necesitado modelos matemáticos más complejos y se han desarrollado más sofisticados algoritmos de control. Sin embargo, el uso de modelos matemáticos muy complejos puede inhibir seriamente la habilidad para desarrollar los algoritmos de control. Afortunadamente, es posible utilizar modelos simples (linealizados) de sistemas para efectuar el diseño. Estos controladores cumplen satisfactoriamente las especificaciones del sistema alrededor de un punto de operación en donde el sistema lineal es válido.

En el mundo real y con las crecientes demandas de los procesos actuales, se encuentran sistemas cuyo modelo matemático no está disponible o podría estarlo pero a un costo y esfuerzo muy elevados. Surgen entonces técnicas diferentes a las convencionales, en donde es preciso tener conocimiento del desempeño del sistema o basarse en expresiones cualitativas de la experiencia de la gente que trabaja con el proceso en cuestión y del mismo ingeniero de control.

Los modernos métodos de producción logrados con equipos dotados de un alto grado de sofisticación permiten mejores resultados en las tareas productivas que no podrían llevarse a cabo mediante las técnicas convencionales de control, son ejemplos de estas características:

- ▣ Cambios frecuentes en las líneas de producción.
- ▣ Introducción de más avanzados métodos de producción.
- ▣ Incremento en la producción y velocidad de la manufactura.
- ▣ La introducción de grandes sistemas de control que integren en un solo sistema las políticas, planes, operación, monitoreo y supervisión de las tareas de control.

En respuesta a estas necesidades resultan problemas de control debidos a:

- ▣ Cambio en las condiciones y en los puntos de operación.
- ▣ No linealidades fuertes y desempeño variante en el tiempo.
- ▣ Inestabilidad inherente al proceso o a alguna parte de él.
- ▣ Datos incompletos o excesivos y ruidosos.
- ▣ Mezcla de conocimiento cualitativo y cuantitativo.

Se debe señalar que cuando se cuente con un modelo matemático que aproxime el comportamiento del sistema satisfactoriamente, es más viable el diseño de un controlador convencional, permitiendo el ahorro de tiempo y dinero. De todas formas es común encontrar controladores inteligentes que mejoran el desempeño de un sistema controlado por métodos convencionales.

A diferencia del control convencional, el control inteligente aún no tiene una trayectoria bien reconocida y fundamentada en donde se permita hacer el análisis de parámetros importantes como la estabilidad, robustez, controlabilidad, observabilidad, etc. Es por eso que el control inteligente ha sido mirado con mucho recelo y considerado como un nuevo paradigma.

1.3.4 Técnicas de control inteligente

1.3.4.1 Sistemas Expertos Un sistema experto es básicamente un conjunto de programas que intentan imitar el proceso de razonamiento humano y el conocimiento de expertos humanos en la solución de problemas específicos.

Típicamente un Sistema Experto (SE) consta de tres componentes principales: base de conocimiento, máquina de inferencia y la interface para el usuario.

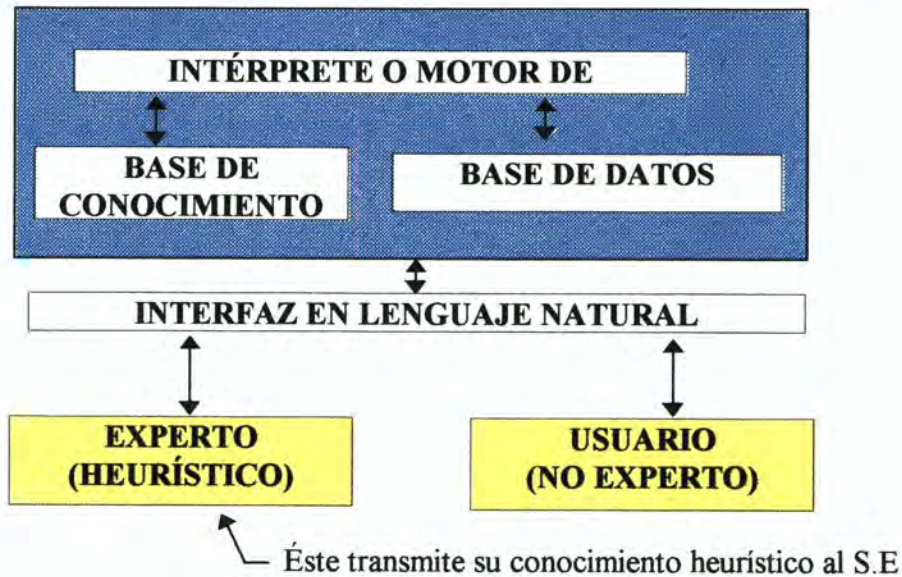


Figura 1-5 Diagrama Estructural de un Sistema Experto

La base de conocimiento contiene el conocimiento necesario para el entendimiento, formulación y solución de problemas. La construcción de esta base de conocimiento depende de un proceso de adquisición del conocimiento bien definido y estructurado, además de ciertas normas de representación del conocimiento.

El motor de inferencias, el cual es realmente el “cerebro” de un SE, provee métodos para el razonamiento acerca de la información de la base de conocimiento y para llegar a conclusiones a partir de éste. Una máquina de conocimiento normalmente consiste en un intérprete que aplica las correspondientes reglas de la base de conocimiento y un planeador que mantiene control sobre ciertas propiedades básicas del sistema.

La interface del usuario debe ser lo más amigable posible para permitir un fácil uso del sistema. Frecuentemente es usado para la comunicación del usuario con el SE un procesador de lenguaje complementado con menús y gráficas.

1.3.4.2 Redes Neuronales (RN)

Las redes neuronales surgieron inicialmente como una forma de simular el comportamiento de los enlaces neuronales de los seres vivos e incluso con la intención de emularlo para llegar a reemplazarlos en casos de necesidad. Básicamente existen dos tipos de redes neuronales: RN modeladas y RN funcionales. En general, cuando se habla de redes neuronales artificiales, se hace referencia a las RN funcionales. Su aplicación en control se basa principalmente en su capacidad de aprendizaje que permite obtener modelos no matemáticos de los sistemas, reconocer patrones de comportamiento o hacer análisis de imágenes, etc.

Una red neuronal artificial está caracterizada por: el patrón de conexión entre neuronas "ARQUITECTURA", el método para determinar los pesos "ENTRENAMIENTO" o "APRENDIZAJE" y por su "FUNCIÓN DE ACTIVACIÓN".

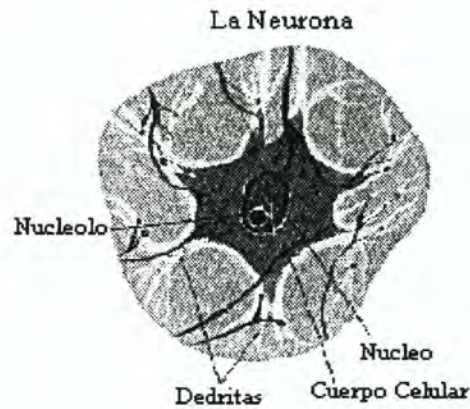


Figura 1-6 Estructura de una neurona

Arquitectura las redes neuronales están conformadas por varias capas de “neuronas”. Una RN consta de una capa de entrada, una de salida y una o varias capas intermedias u ocultas. En la figura 1-6 se muestra la estructura de una neurona y en la figura 1-7 una red neuronal.

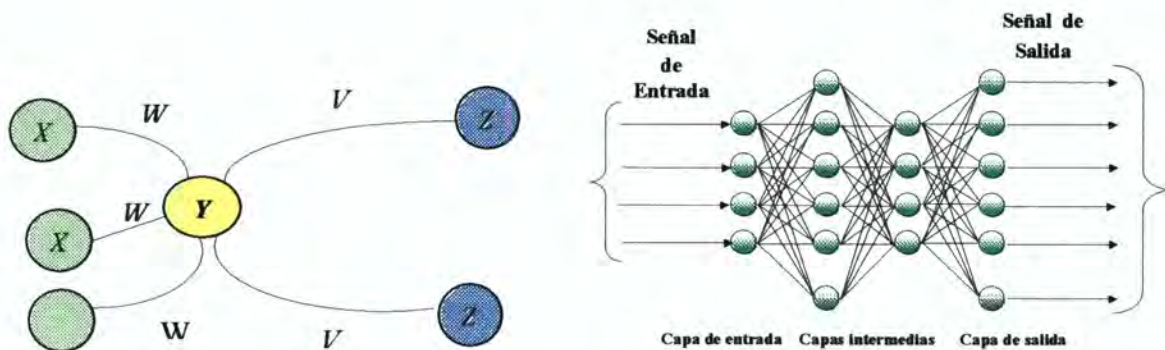


Figura 1-7 Red neuronal artificial

Función de activación: de acuerdo con la sumatoria de las entradas ponderadas, esta función determina el nivel de salida. Las funciones más comunes son : signo, identidad y sigmoide, esta última es la de mayor difusión.

Aprendizaje las RN pueden ser entrenadas con base en una serie de patrones de prueba conocidos (aprendizaje supervisado) o con base en criterios permanentes de evaluación (aprendizaje no supervisado). Dentro de estos grupos existen muchos métodos y algoritmos de entrenamiento, el más sencillo y difundido es el de retropropagación del error (Backpropagation).

Entrenamiento de redes neuronales mediante retropropagación:

- ❑ Primero, la red neuronal es inicializada con ciertos pesos, generalmente aleatorios.
- ❑ Luego se aplican las entradas para uno de los patrones de prueba; las salidas obtenidas de la red se comparan con las deseadas, obteniendo de esta forma un error.
- ❑ Posteriormente se ajustan los pesos de conexión entre las neuronas de la capa de salida y la capa anterior de acuerdo con el gradiente del error (variación con respecto al patrón anterior).
- ❑ El gradiente obtenido se utiliza como el error para el ajuste de las conexiones entre las siguientes dos capas (de tal forma que el error se propaga hacia atrás).

Se repiten luego los últimos tres pasos para cada patrón de prueba y se continúa el entrenamiento hasta que se alcance una condición de parada para el algoritmo. Generalmente se fija un umbral para el error promedio (de todos los patrones) y una vez superado se detiene el entrenamiento.

Redes Neuronales en control las aplicaciones habituales de las RN en control giran alrededor de:

- ❑ Identificación de patrones (para identificar sistemas, clasificar condiciones, procesamiento de imágenes, etc.)
- ❑ Aprendizaje de modelos dinámicos directos (simulación de plantas) o inversos (control en cascada).
- ❑ Como mecanismo de adaptación para controladores convencionales.

1.3.4.3 Algoritmos Genéticos los principios básicos de Algoritmos Genéticos (AG) fueron propuestos por J. Holland en 1975. Un AG es esencialmente un procedimiento de búsqueda, modelado sobre la mecánica de la selección natural de las especies (no en un proceso de razonamiento simulado).

Los AG están constituidos principalmente por los siguientes elementos :

- ❑ *Organismo*: es una solución candidata, con conocimiento empotrado en su representación abstracta.
- ❑ *Población*: es un conjunto de organismos que coexisten.
- ❑ *Generaciones*: son poblaciones sucesivas cronológicamente.

En general, un AG actúa sobre una población de individuos u organismos mediante la aplicación de tres operadores genéticos: reproducción, cruzamiento y mutación.

El proceso de **reproducción** copia cromosomas desde una generación a la siguiente, basándose en sus valores adecuados. La probabilidad de repetir un cromosoma específico sobre el siguiente es proporcional a su desempeño o valor adecuado. Como su nombre lo indica, el **cruzamiento** cruza pares de cromosoma aleatoriamente seleccionados con el fin de producir nuevos descendientes. Los cromosomas reproducidos contienen materiales o información de sus padres originales. La **mutación** actúa sobre un cromosoma cambiándole aleatoriamente partes de sus representación. El objetivo de la mutación es el de rescatar un buen individuo que se pudiera haber perdido debido a posibles reproducciones y cruzamientos adversos.

La estructura básica de un AG sugerida por Holland es la siguiente :

Algoritmo Genético Simple

```
[
  inicializar población;
  evaluar población;
  mientras que el criterio de parada no se alcance
  [
    seleccione soluciones para la siguiente población;
    realice cruzamiento y mutación;
    evalúe población;
  ]
]
```

Características de los AG :

- ❑ Optimizan el compromiso entre buscar nuevas opciones y utilizar la información obtenida hasta el momento.
- ❑ Poseen la capacidad de paralelismo implícito: búsqueda extensiva sin evaluar todas las opciones.

- ❑ Son algoritmos aleatorizados, esto es, usan operadores cuyos resultados son gobernados por la probabilidad.

Entre las aplicaciones más comunes de los AG se encuentran la identificación de parámetros y de sistemas, control robusto, sistemas de navegación autónomos, reconocimiento de patrones y de voz. En los chips fabricados bajo la tecnología VLSI permiten la optimización de celdas y de rutas, reduciendo el tamaño del chip.

1.3.4.4 Lógica Difusa la lógica difusa se ha convertido en una de las herramientas más populares en la industria, en especial en la japonesa. Inicialmente fue introducida por Zadeh como la teoría de “Conjuntos Difusos” en 1965. Los conjuntos difusos son una generalización de la teoría de conjuntos convencional y contienen objetos que pertenecen imprecisamente al conjunto. El grado de membresía al conjunto está definido por el valor de una función de pertenencia, la cual toma valores entre 0 y 1. Esta técnica permite una forma de representar las incertidumbres de un modelo matemático.

La lógica difusa no es un método estadístico en donde sean representados eventos de carácter aleatorios más bien, intenta capturar y cuantificar imprecisión no aleatoria. Puede entenderse como una herramienta útil para construir modelos de razonamiento humano que reflejen el carácter vago y cualitativo que éste tiene. El enorme interés despertado por la lógica difusa se basa en la posibilidad de tratar problemas demasiado complejos o muy mal definidos como para admitir un tratamiento por los métodos convencionales.

La lógica difusa fue aplicada por primera vez para aspectos de control por Mamdani en 1974. Mientras la evolución del control difuso fue muy controvertida dentro de la comunidad de la ingeniería de control, los japoneses han implementado con éxito diversas aplicaciones, tales como lavadoras, cámaras de video, aire acondicionado, sistemas de navegación para barcos, control de un tren subterráneo, transmisión de velocidades en automóviles, entre otros.

Es recomendable la aplicación de Control Difuso cuando :

- ❑ Se está tratando con procesos muy complejos y su modelo matemático no es lo suficientemente simple.
- ❑ Se tienen procesos con no-linealidades importantes.
- ❑ El procesamiento del conocimiento (lingüísticamente formulado) de un experto va a ser realizado.

No sería buena idea aplicar control difuso si :

- ❑ La teoría de control convencional brinda un resultado satisfactorio o cuando se tiene un modelo matemático del sistema fácilmente resoluble o si el problema no es solucionable.

2. INTRODUCCIÓN A LA LÓGICA DIFUSA

La lógica difusa nace en 1965 a partir de la publicación del artículo “Fuzzy Sets” escrito por Lofti Zadeh para la revista Information and Control. Según Zadeh, la lógica difusa permite aprovechar la “imprecisión” con la cual razonamos los seres humanos y surge como una lógica multivaluada llegando a ser más general que la tradicional lógica clásica de unos y ceros (bivaluada). En este capítulo se hace un recorrido histórico sobre los pensamientos que preceden al surgimiento de la lógica difusa y que atañen de alguna manera con ésta, luego se describe la teoría de los conjuntos difusos y sus operaciones tal y como lo hizo Zadeh hace 35 años.

2.1. DESARROLLO HISTÓRICO

Desde la época de los grandes filósofos griegos se ha venido cuestionando la efectividad de la dicotomía *cierto-falso* y posteriores pensadores también han formulado sus conceptos hasta cuando Zadeh habla de los conjuntos difusos y moldea luego la teoría de la lógica difusa como se observa a continuación:

380 a.C. Aristóteles propone la existencia de grados de verdad o falsedad.

Siglo XVIII En Inglaterra el filósofo David Hume habla de la lógica del sentido común (razonamiento basado en la experiencia que la gente comúnmente adquiere de sus vivencias por el mundo).

El filósofo norteamericano Charles Sander Pierce, fue el primero en considerar la vaguedad en vez de la dicotomía cierto-falso, como una forma de enmarcar cómo el mundo y las personas funcionan.

También en este siglo es inventada la teoría original de conjuntos clásicos de unos y ceros por el matemático alemán Georg Kantor.

El famoso filósofo y matemático inglés Bertrand Russell estudió la vaguedad del lenguaje, así como su precisión.

1920 El filósofo polaco Jan Lukasiewicz propone la primera lógica de vaguedad. Desarrolló conjuntos con posibles valores de pertenencia 0, $\frac{1}{2}$ y 1 (lógica trivaluada). Posteriormente los extendió hacia un número infinito de valores entre 0 y 1 (lógica multivaluada).

1962 Lotfi Zadeh cuestiona la efectividad de las matemáticas tradicionales, las cuales resultaban intolerantes ante la imprecisión y ante verdades parciales.

1964 Aparece por primera vez la noción de conjuntos difusos en un memorándum debido al mismo Zadeh en la Universidad de California en Berkeley. Dicho memorándum es publicado un año más tarde bajo el título: "Fuzzy Sets" (Conjuntos difusos)⁶.

Después de la aparición de este artículo, los conjuntos difusos se convirtieron en una herramienta de gran utilidad para describir conceptos vagos e imprecisos que emanan del razonamiento humano, permitiendo describir y resolver problemas de planteamientos vagos y que por tanto no tenían solución dentro de las teorías convencionales.

En sucesivos trabajos desde 1971 Zadeh produce los elementos formales que terminaron componiendo el cuerpo de la doctrina de la lógica difusa tal y como se conoce en la actualidad. Así en :

1971 Propone el empleo de conjuntos difusos para manejar cuantitativamente conceptos cualitativos

1972 Propone los llamados modificadores lingüísticos que se emplean en el lenguaje natural para modificar los adjetivos.

⁶ ZADEH Lotfi A. "Fuzzy Sets." En Information and Control, Vol. 8, (p. 338-353) 1965.

- 1973** Plantea el Principio de Incompatibilidad, la Regla Composicional de Inferencia y propone también las proposiciones condicionales Si-Entonces (If-Then) para describir las relaciones Entrada/Salida de sistemas complejos.
- 1975** Desarrolla el concepto de variable lingüística y le da forma definitiva a la Regla Composicional de Inferencia.
- 1978** Publica un artículo sobre la teoría de posibilidades y la Lógica Difusa.
- 1979** En sendos artículos describe una elaborada teoría del Razonamiento Aproximado.
- 1989** Representación de conocimiento en Lógica Difusa Zadeh.
- 1994** Lógica difusa, Redes Neuronales y Soft Computing Zadeh.

Razonamiento Aproximado la lógica difusa, en el sentido de teoría de conjuntos difusos da las bases para el diseño de modelos difusos. La teoría de conjunto difusos y la amplia área de la misma lógica difusa no es un método específico para cualquier aplicación más aún que la lógica Booleana o la probabilidad, es una lógica para un tipo particular de aplicaciones. La teoría de conjuntos difusos soporta una teoría más general que la lógica difusa para el cálculo de sentencias difusas. Ésta en cambio, soporta la construcciones lógicas usadas para

crear y manipular sistemas difusos. Tiene el nombre más general de razonamiento aproximado. La Figura 2.1 muestra cómo están relacionadas estas plataformas.

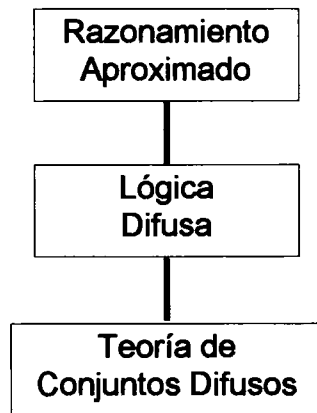


Figura 2-1 Relación entre plataformas

Frecuentemente lógica difusa y el razonamiento aproximado son usados indistintamente para indicar el proceso de expresión de procesos imprecisos o aproximados y sus relaciones. A pesar de su cercanía se debe saber que la lógica difusa es una representación más formal de la teoría de conjuntos difusos, y el razonamiento aproximado como un método de codificar el conocimiento a través de reglas difusas condicionales e incondicionales, el razonamiento aproximado maneja no sólo las matemáticas fundamentales de la lógica difusa sino que también incorpora un conjunto de heurísticas que se encuentran fuera de la teoría de conjuntos difusos para trabajar consistentemente y predeciblemente. El uso de los cualificadores es un ejemplo de un concepto que atañe a la plataforma del razonamiento aproximado pero no a la lógica difusa o a la teoría de conjuntos difusos.

2.2 CONJUNTOS DIFUSOS

Según Zadeh en⁷ *“Un conjunto difuso es una clase de objetos con un continuo grado de pertenencia. Tal conjunto está caracterizado por una función de pertenencia (característica) la cual asigna a cada objeto un grado de pertenencia entre el intervalo cero y uno. Las nociones de inclusión, unión, intersección, complemento, relación, inversamente, etc., son extendidos a tales conjuntos, y varias propiedades de estas nociones son establecidas en el contexto de los conjuntos difusos”*. Esta definición fue la que le dio inicio a la teoría de conjuntos difusos y abrió las puertas a la lógica difusa.

Para fines prácticos podemos definir los conjuntos difusos de la siguiente forma⁸: Sea U Una colección de objetos denotados generalmente por $\{u\}$, el cual puede ser discreto o continuo. U es llamado el universo de discurso y u representa un elemento genérico de U .

Conjunto Difuso: Un conjunto difuso A en un universo de discurso U está caracterizado por una función de pertenencia μ_A la cual toma valores en el intervalo $[0,1]$ definida $\mu_A : U \rightarrow [0,1]$.

⁷ ZADEH, Op. Cit., p. 341

⁸ LEE, Chuen C. "Fuzzy Systems: Logic in Controller. Parte I y II." En IEEE Transaction on Systems Man. And Cybernetics, Vol. 20, No. 2 (p. 404-435) 1990

Un conjunto difuso puede verse como una generalización del concepto de un conjunto ordinario o clásico cuya función de pertenencia sólo puede tomar uno de dos valores $\{0,1\}$.

Así, un conjunto difuso A en U puede ser representado como un conjunto de pares ordenados con un elemento genérico u y su grado de pertenencia $\mu_A(u)$:

$$A = \{(u, \mu_A(u)) | u \in U\}. \quad (2.0)$$

Cuando U es continuo, un conjunto difuso A puede escribirse como

$$A = \int_U \mu_A(u) / u. \quad (2.1)$$

En el caso de que el universo de discurso sea discreto A se puede representar como

$$A = \sum_{i=1}^n \mu_A(u_i) / u_i. \quad (2.2)$$

Toda la información de un conjunto difuso está contenida en su función de pertenencia $\mu(x)$.

Las siguientes definiciones nos permiten comprender mejor el concepto de función de pertenencia $\mu(x)$.

Soporte: el soporte o sustento de un conjunto difuso A es el conjunto clásico de todos los puntos $u \in U$ tal que, su función de pertenencia sea $\mu_A > 0$.

Centro: de un conjunto difuso A es el conjunto de puntos o punto $u \in U$ para el que su función de pertenencia μ_A alcanza su máximo valor.

Singleton difuso: si el soporte de un conjunto difuso A es un simple punto en U para el cual $\mu_A = 1$, el conjunto difuso es llamado un singleton difuso.

Cruce: el elemento $u \in U$ para el cual $\mu(x) = 0.5$ es llamado punto de cruce.

Límite: se define por el conjunto de puntos $u \in U$ para el que la función de pertenencia es $0 < \mu_A < 1$.

Altura: se define como el máximo valor de la función de pertenencia $\max\{\mu_A(x)\}$.

Conjunto difuso normal: aquel que tiene al menos un elemento en el universo con pertenencia igual a 1.

Una gráfica que ilustra mejor las definiciones anteriores para un conjunto difuso con función de pertenencia triangular, se muestra en la figura 2-2.

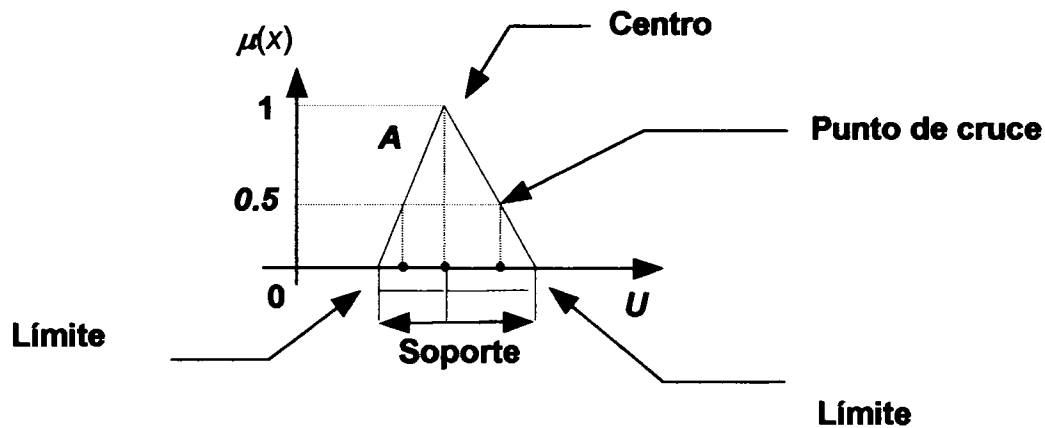


Figura 2-2 Función de pertenencia del conjunto difuso A

2.3 OPERACIONES ENTRE CONJUNTOS DIFUSOS

Al igual en que los conjuntos clásicos, existen una serie de operaciones para los conjuntos difusos, pero antes de hablar de las operaciones hagamos algunas definiciones: sean A , B y C conjuntos difusos definidos en el universo de discurso U y con funciones de pertenencia μ_A , μ_B y μ_C respectivamente,

- ▣ Un conjunto difuso es *vacío* si y sólo si su función de pertenencia es idénticamente cero en U .
- ▣ Dos conjuntos difusos A y B son iguales, escrito como $A = B$, si y sólo si $\mu_A(x) = \mu_B(x)$ para todos los x en U (otra forma más simplificada $\mu_A = \mu_B$).

Operaciones la teoría de operaciones entre conjuntos difusos se define básicamente a través de sus funciones de pertenencia de la siguiente forma:

□ Unión. La unión ($A \cup B$) de dos conjuntos difusos A y B con sus respectivas funciones de pertenencia μ_A y μ_B es un conjunto difuso C , escrito como $C = A \cup B$, cuya función de pertenencia esta definida para todo $x \in U$ por

$$\mu_{A \cup B}(x) = \max \{ \mu_A(x), \mu_B(x) \} \quad (2.3)$$

o, de forma abreviada

$$\mu_C = \mu_A \vee \mu_B$$

En la gráfica 2-3 podemos ver los conjuntos difusos A y B y su respectiva unión

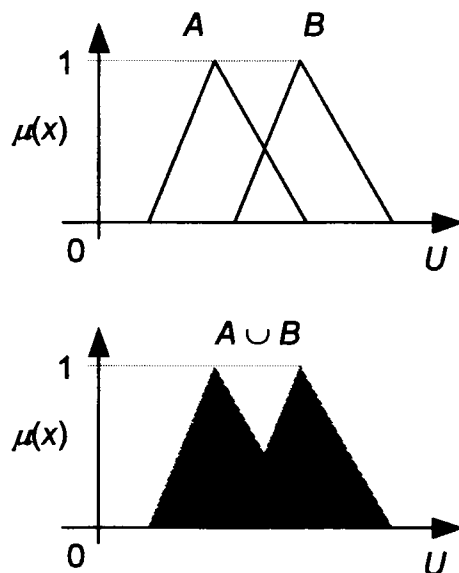


Figura 2-3 Representación gráfica de la unión de A y B

La unión de A y B se puede definir como el conjunto más pequeño contenido en ambos A y B . Más precisamente, si C es algún conjunto difuso que contiene a ambos A y B , éste también contiene la unión de A y B .

□ La intersección ($A \cap B$) de dos conjuntos difusos A y B es un conjunto difuso en U con función de pertenencia definida para todo $x \in U$ como

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} \quad (2.4)$$

o, de forma abreviada

$$\mu_C = \mu_A \wedge \mu_B$$

$$\mu_C = \mu_A \wedge \mu_B$$

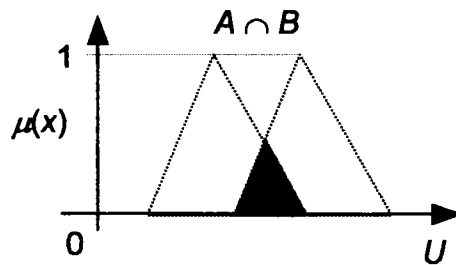


Figura 2-4 Representación gráfica de la intersección de A y B

El complemento de un conjunto difuso A está denotado por \overline{A} y se define como

$$\mu_{\overline{A}} = 1 - \mu_A \quad (2.5)$$

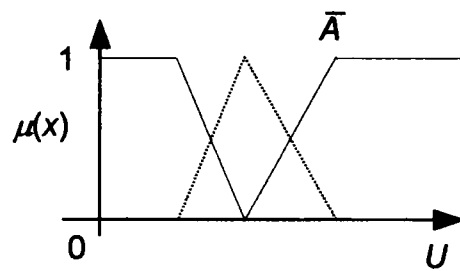


Figura 2-5 Representación gráfica del complemento (\bar{A}) de A

▣ **Contenido.** A está contenido en B (o equivalentemente, A es un subconjunto de B , o A es más pequeño o igual que B) si y sólo si $\mu_A \leq \mu_B$ para todo $x \in U$. En símbolos

$$A \subset B \Leftrightarrow \mu_A \leq \mu_B \quad (2.6)$$

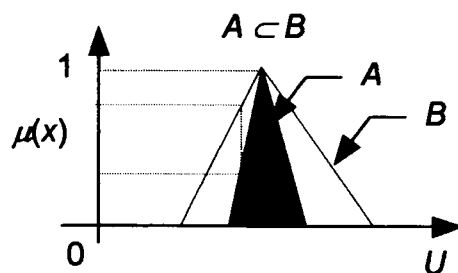


Figura 2-6 Representación gráfica de $A \subset B$

3. SISTEMAS DIFUSOS

3.1. INTRODUCCIÓN

Dentro de la literatura podemos destacar la definición que hace Wang⁹ de sistemas difusos como, "*Un sistema de lógica difusa o difuso, es el nombre dado a los sistemas que tienen una relación directa con conceptos difusos (como conjuntos difusos, variables lingüística, etc.) y lógica difusa*". Wang describe los más populares sistemas de lógica difusa que existen y los clasifica en tres tipos: sistemas de lógica difusa puros, sistemas Takagi y Sugeno's, y sistemas de lógica difusa con fuzzificación y defuzzificación. Debemos hacer claridad que este último tipo de sistemas es conocido también como sistemas de tipo Mamdani ya que fue él quien primero propuso este tipo de sistemas. Veamos a continuación estos tres tipos de sistemas.

Sistemas de lógica difusa puros (SLDP) la configuración básica se puede ver en la figura 3.1. Donde la base de reglas difusas consta de una colección de reglas difusas IF-THEN, y

⁹ WANG, Li-Xin., En "Adaptative Fuzzy Systems and Stability Analysis."

el motor de inferencia difusa usa estas reglas difusas IF-THEN para determinar un mapeo desde conjuntos difusos en el universo de discurso de entrada $U \subset R^n$ a conjuntos difusos en

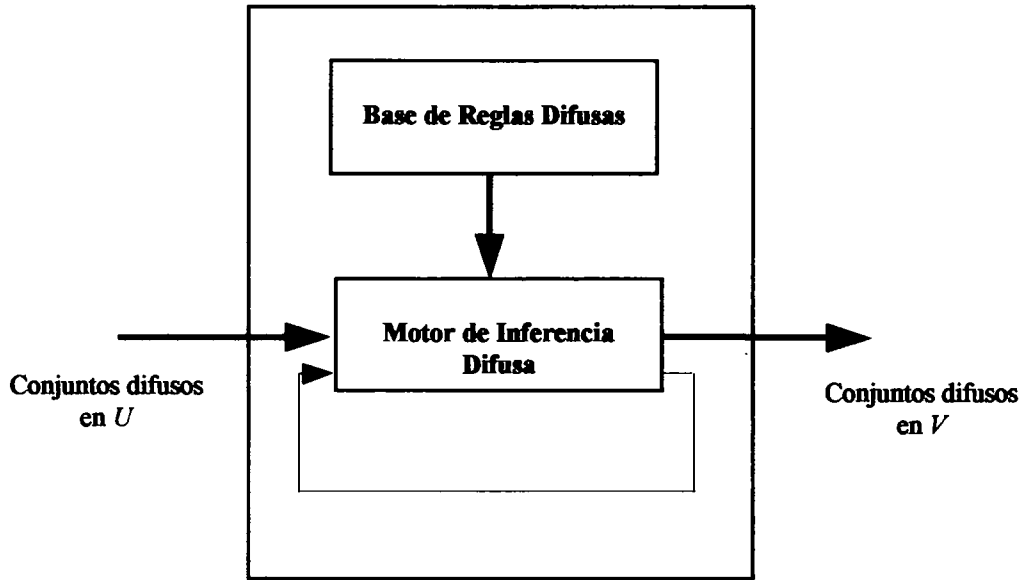


Figura 3-1 Configuración básica de un sistema de lógica difusa puro.

el universo de discurso de salida $V \subset R$ basándose en principios de lógica difusa. Las reglas difusas son de la siguiente forma:

$$R^{(l)} : \text{IF } x_1 \text{ es } F_1^l \text{ y } \dots \text{ y } x_n \text{ es } F_n^l, \text{ THEN } y \text{ es } G^l \quad (3.1)$$

donde F_i^l y G^l son conjuntos difusos, $\underline{x} = (x_1, \dots, x_n)^T \in U$ y $y \in V$ son variables lingüísticas de entradas y salidas respectivamente, y $l = 1, 2, \dots, M$. Cada regla define un conjunto difuso $F_1^l \times \dots \times F_n^l \rightarrow G^l$ en el producto espacial $U \times V$. El principio de lógica difusa más usado en el motor de inferencia difusa es el llamado composición sup-star. Específicamente, sea A un conjunto arbitrario en U , y A' la entrada al SLDP de la figura 3-1, entonces la

salida es determinada por cada regla difusa IF-THEN como un conjunto difuso A o $R^{(j)}$ en V cuya función de pertenencia es:

$$\mu_{A \circ R^{(j)}}(y) = \sup_{\underline{x} \in U} \left[\mu_A(\underline{x}) \otimes \mu_{F_1^I \times \dots \times F_n^I \rightarrow G^I}(\underline{x}, y) \right] \quad (3.2)$$

Donde el operador " \otimes " es un operador *T-norma*, $\mu_A(x)$ es usado para representar la función de pertenencia del conjunto difuso A . La salida final del SLDP es un conjunto difuso A o $(R^{(1)}, \dots, R^{(M)})$ en V que es una combinación de la los M conjuntos difusos de $\mu_{A \circ R^{(j)}}(y)$; que es,

$$\mu_{A \circ (R^{(1)}, \dots, R^{(M)})}(y) = \mu_{A \circ R^{(1)}}(y) \oplus \dots \oplus \mu_{A \circ R^{(M)}}(y) \quad (3.3)$$

donde el operador " \oplus " es un operador *T-conorma*. Si el sistema se realimenta como muestra la figura 3-1 tendremos el llamado sistema difuso dinámico, que es un SLDP donde las entradas dependen de sus salidas.

Los SLDP constituyen una parte esencial de los sistemas de lógica difusa. Estos son una estructura general en la cual la información lingüística de humanos expertos es cuantificada y los principios de la lógica difusa son utilizados para hacer un uso sistemático de la información lingüística. Una desventaja principal de los SLDP es que las entradas y las salidas son conjuntos difusos, mientras que en muchos sistemas de ingeniería las entradas y salidas son variables con valores reales. Para cubrir esta desventaja Takagi y Sugeno,

propusieron en 1985 un sistema de lógica difusa con entradas y salidas que tienen variables con valores reales.

Sistemas difusos Takagi y Sugeno. En lugar de considerar las reglas difusas IF-THEN de la forma (3.1) de los SLDP, Takagi y Sugeno propusieron el uso de las siguientes reglas difusas:

$$\begin{aligned} L^{(l)} \text{ IF } x_1 \text{ es } F_1^l \text{ and } \dots \text{ and } x_n \text{ es } F_n^l, \\ \text{THEN } y^l = c_0^l + c_1^l x_1 + \dots + c_n^l x_n \end{aligned} \quad (3.4)$$

Donde F_i^l son conjuntos difusos, c_i son parámetros de valores reales, y^l es la salida del sistema debido a la regla $L^{(l)}$, y $l = 1, 2, \dots, M$. Se consideran reglas cuya parte IF (antecedente) es difusa pero la parte THEN (consecuente) es crisp y la salida es una combinación lineal de las variables de entrada. Para un vector de entrada de valores reales $\underline{x} = (x_1, \dots, x_n)^T$, la salida $y(\underline{x})$ del sistema difuso de Takagi y Sugeno's es un peso promediado de los y^l :

$$y(\underline{x}) = \frac{\sum_{l=1}^M \omega^l y^l}{\sum_{l=1}^M \omega^l} \quad (3.5)$$

donde el peso ω^l implica todos los valores de verdad de la premisa de la regla $L^{(l)}$ para la entrada y es calculada como:

$$\omega^l = \prod_{i=1}^n \mu_{F_i^l}(x_i) \quad (3.6)$$

La configuración del sistema Takagi y Sugeno's se muestra en la figura 3-2.

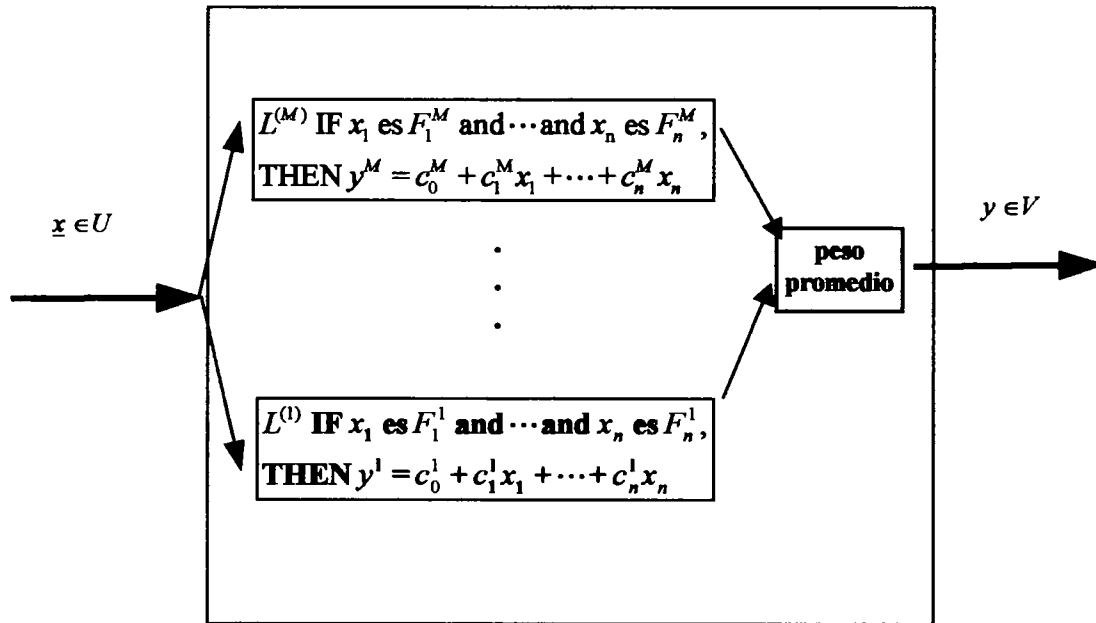


Figura 3-2 Configuración de un sistema Takagi y Sugeno's

Estos sistemas se han aplicado con buen éxito a muchos problemas prácticos. La ventaja de estos sistemas es que proveen una ecuación (3.5) compacta del sistema y por eso, la estimación de parámetros y métodos ordenados de determinación pueden ser desarrollados para estimar los parámetros c_i^l y el orden M . Un punto débil de este sistema es que la parte THEN (consecuente) de la regla no es difusa; por eso, este no provee una estructura natural para incorporar reglas difusas de expertos humanos. Ya que, la regla (3.4) es una

modificación de la regla difusa pura (3.1) (por ejemplo cambia " y es G^i " a " $y = c_o^i$ ", que es el centro de μ_{G^i}).

Sistemas difusos con fuzzificación y defuzzificación (Mamdani). Para usar el sistema SLDP de la figura 3-1 en sistemas de ingeniería donde las entradas y las salidas son variables que toman valores reales, la mejor manera es agregar un fuzzificador a la entrada y un defuzzificador a la salida del SLDP. La configuración básica la podemos ver en la figura 3-3.

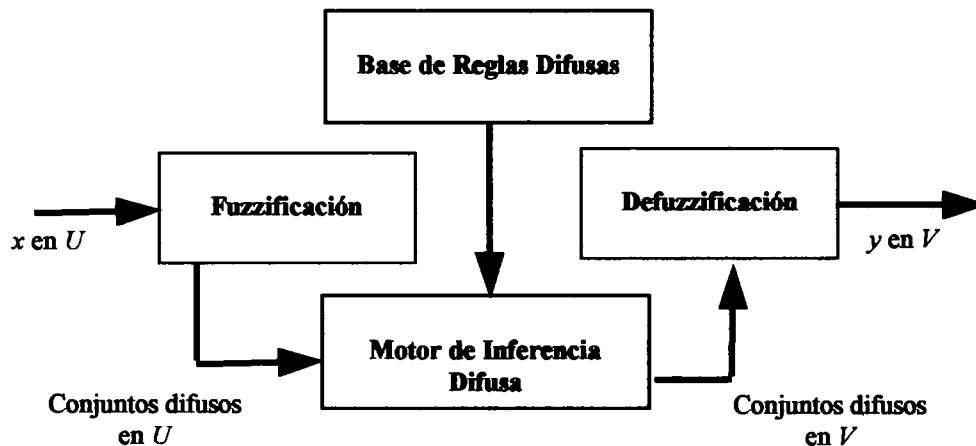


Figura 3-3. Configuración básica de sistema de lógica difusa con fuzzificación y defuzzificación

El fuzzificador mapea puntos crisp en U a conjuntos difusos en U , y el defuzzificador mapea conjunto difusos en V a puntos crisp en V . La base de reglas y el motor de inferencia difusa son los mismo que en los SLDP. En la literatura estos sistemas son llamados a menudo controladores de lógica difusa, debido a que son principalmente utilizados como tales. Esta forma fue propuesta primero por Ebrahim Mandani en 1974 para controlar una máquina de

vapor y ha sido muy bien aplicada en una variedad de procesos industriales y productos de consumo.

Estos sistemas ofrecen muchas cualidades atractivas. Son apropiados para sistemas de ingeniería debido a que sus entradas y salidas son variables que toman valores reales. Además proveen una estructura natural para incorporar reglas difusas de humanos expertos. Hay mucha libertad en las opciones de fuzzificación, motor de inferencia y defuzzificación, también podemos obtener sistemas difusos satisfactorios para un problema en particular. Podemos desarrollar diferentes algoritmos de entrenamiento para estos sistemas de lógica difusa así que estos proveen una efectiva estructura para integrar información lingüística y numérica.

3.2 VARIABLE LINGÜÍSTICA

Existen dos interpretaciones para el concepto de variable lingüística; formalmente una variable lingüística es definida de la siguiente manera¹⁰:

Definición formal, *Variable lingüística*: Una variable lingüística está caracterizada por una quintupleta $(x, T(x), U, G, M)$ en donde x es el nombre de la variable lingüística; $T(x)$ es el conjunto de términos de x , que son, el conjunto de nombres de valores lingüísticos de x con

¹⁰ LEE, Op. Cit., p. 404-435

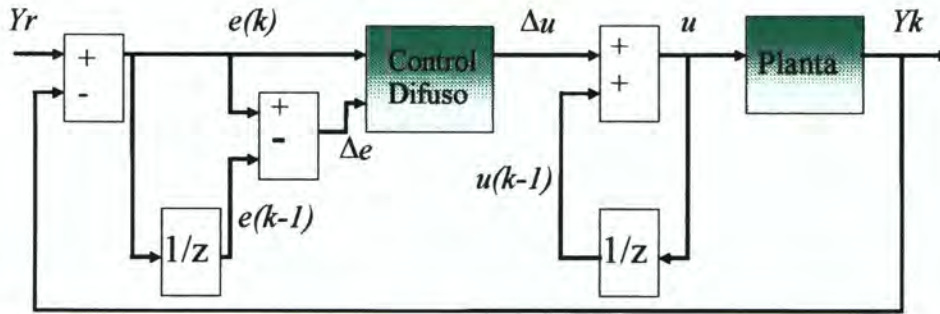


Figura 4-6 Controlador PI difuso.

4.5 CONTROL DIFUSO POR MODOS DESLIZANTES

La tecnología actual involucra sistemas que son altamente no lineales. Variantes en el tiempo y con ambientes inciertos. El control más eficiente y óptimo de esta clase de sistemas se ha convertido en una amplia área de investigación. Los esquemas tradicionales de control presentan severas deficiencias cuando se aplican a tales sistemas.

El Control de Estructura Variable (CEV) que fue desarrollado por investigadores rusos en la década de los 50's, ha sido ampliamente estudiado como un método que trata con este tipo de sistemas. La carencia de un procedimiento de diseño, así como el fenómeno de "chattering" (movimiento oscilatorio de altas frecuencias) impidieron que los Sistemas de CEV tuvieran gran aceptación hasta los años 70's. Fue en esta década cuando los sistemas de CEV recibieron atención debido a su comprobada robustez e invarianza. Desde entonces muchas investigaciones se han conducido para desarrollar procedimientos que permitan reducir el "chattering".

Los sistemas de control realimentados tradicionales constan de la planta que va a ser utilizada mediante un simple controlador, mientras que los sistemas de CEV contienen múltiples estructuras de realimentación, donde solamente un ciclo de control está activo en determinado instante. El ciclo de control activo queda determinado por una función de conmutación.

“Los sistemas de control por modos deslizantes¹⁵ (SCMD) son un tipo particular de los sistemas de CEV, en los cuales la trayectoria del estado es obligada a dirigirse hacia una superficie dada y a “deslizar” por ella y de esta manera alcanzar el punto de operación deseado. Teóricamente, los elementos de conmutación del SCMD deberían operar a unas frecuencias infinitas para producir un modo deslizante ideal. Es por esto, que en la práctica se produce el “chattering” alrededor de la superficie de deslizamiento” Nota.

Mediante la introducción de un sistema de control por modos deslizantes difuso (SCMDD) es posible contar con las bondades del controlador difuso y las del control por modos deslizantes. Fuzzificando la superficie de deslizamiento, el SCMDD produce un modo deslizante difuso, similar al modo deslizante convencional, pero con mucho menos “chattering”. El sistema en lazo cerrado, podrá estar caracterizado por una superficie de deslizamiento definida por el diseñador y las reglas difusas se pueden obtener fácilmente.

¹⁵ NARANJO P. Freddy., En “Control de estructura variable en tiempo discreto : comparación de algoritmos”. 7º Congreso Latinoamericano de control automático. LACC-IFAC. Buenos Aires 1996. P 1040-1045.

En la elaboración de las reglas lingüísticas de control, se debe tener en cuenta (en forma similar que con los SCMD) que por encima de la línea de conmutación, $S=0$, se genere una salida de control negativa y por debajo una positiva.

Una técnica general para el diseño es la partición del plano de fase en dos semiplanos por medio de la línea de conmutación. Dentro de los dos semiplanos se producen las salidas de control positivas y negativas respectivamente, su magnitud dependerá de la distancia entre el vector de estados y la línea de conmutación.

Una desventaja de este método son los cambios drásticos de la variable manipulada, lo cual produce oscilaciones sobre la planta que está siendo controlada. Esto puede ser evitado por medio de una capa límite cercana a la línea de conmutación, la cual suaviza la señal de control y asegura que el estado del sistema se mantendrá dentro de dicha capa. Tal región se puede conseguir mediante una función de saturación.

Un procedimiento sencillo pero bastante utilizado para el diseño de los SCMD se puede describir de la siguiente manera:

Se define el error entre la salida del sistema $y(kT)$ y la señal de referencia $r(kT)$ como

$$e1(kT) = r(kT) - y(kT) \quad (1)$$

$$\text{y} \quad de1(kT) = e2(kT) \quad (2)$$

La superficie de deslizamiento se define como

$$S = e_2(kT) + c_1 e_1(kT) \quad (3)$$

De igual manera
$$dS = \{ S(kT) - S((k-1)T) \} / T$$

$$= c_1 e_2(kT) - f(e_1(kT)) - g(e_1(kT))u(kT) - d(kT)$$

Se asume que $g(e_1(kT))$ es mayor que cero y $d(kT)$ es la señal de perturbación, por lo tanto realizando el producto SdS y haciendo que $S = 0$, se puede hallar el valor de c_1 .

Sean
$$V(x(t)) = \frac{1}{2} S^2 \text{ y}$$

$$dV(x(t)) = SdS$$

Con el fin de satisfacer el teorema de estabilidad de Liapunov ($SdS < 0$), se definen el conjunto de reglas. Un ejemplo podría ser:

“Si S y dS son Positivo Grande Entonces u es Positivo Grande”

Las reglas lingüísticas de control se pueden determinar intuitivamente mediante un análisis de la ecuación del producto SdS . Si se sabe que SdS es Positivo Grande, la señal de control deberá ser también Positivo Grande para hacer que SdS decrezca rápidamente, de modo que los requerimientos para la condición de estabilidad $SdS < 0$ puedan satisfacerse.

En la figura 4-7, se muestra la trayectoria del estado (e vs de) alcanzando la línea de deslizamiento y deslizándose por ésta dirigiéndose hacia el origen.

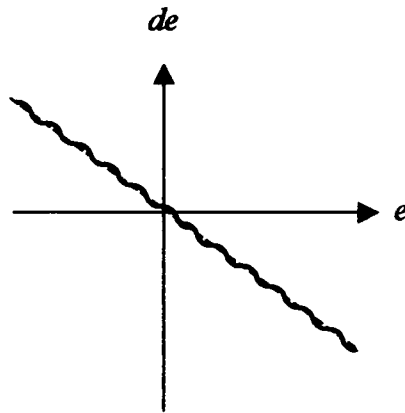


Figura 4-7 Trayectoria del estado e vs de .

4.6. CONTROL DIFUSO ADAPTATIVO

Podemos definir adaptación¹⁶ como *"la aptitud de un sistema para acomodar (acostumbrar, adaptar) su comportamiento a cambios en el medio."* Cuando estamos modelando una estrategia de control y se define un controlador es posible encontrar parámetros para el controlador que permitan de manera satisfactoria dar solución al problema de control específico, pero debido a la dinámica de los sistemas es posible que los parámetros deban ser modificados ante cambios en el medio, de ahí surge la necesidad de controladores con capacidad de adaptación. Un ejemplo de esta adaptación es podría presentar en la necesidad de ajuste de los parámetros de sintonía de un PID.

¹⁶ HABER, Rodolfo., *"Curso Introduccion al Control Difuso (Fuzzy)"* CUAO, Centro de Investigaciones, Marzo de 1995 Cali.

En el caso de la lógica difusa esta posibilidad se hace más evidente ya que existe la opción que la base de conocimiento requiera de la adaptación para implementar una correcta estrategia de control lingüística.

Debemos entender que la adaptación de un controlador es un procedimiento que identifica el entorno o vecindad al punto de control, y a partir de allí genera una acción de control sólo para ese entorno (ajuste de parámetros). Siendo cada situación diferente a la actual una situación nueva, de tal manera que si el sistema se encuentra en un punto A de operación con parámetros de operación para el controlador P_A , y este cambia al punto de operación B , el controlador debe realizar la adaptación de sus parámetros a P_B . Si el sistema retorna a A el controlador debe realizar nuevamente la adaptación, esto debido a que el controlador adaptable no tiene memoria, lo que lo diferencia de un control inteligente que reconocería inmediatamente el punto A . Decimos entonces que un controlador adaptable no es inteligente por la ausencia de aprendizaje.

Para un sistema en lazo cerrado podemos definir las siguientes variables: \underline{R} como el vector de referencia del sistema, \underline{Y} el vector de salida del sistema, \underline{Z} el vector de las perturbaciones que afectan el sistema y \underline{P} que recoge los parámetros del controlador veamos el siguiente esquema:

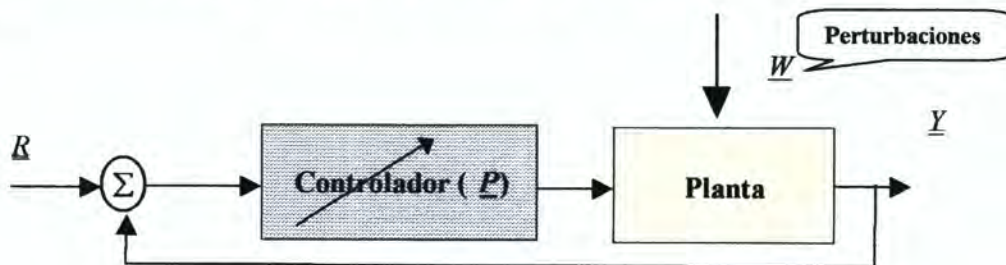


Figura 4-8 Esquema de un controlador adaptable

En el caso de la lógica difusa esta posibilidad se hace más evidente ya que existe la opción que la base de conocimiento requiera de la adaptación para implementar una correcta estrategia de control lingüística.

Debemos entender que la adaptación de un controlador es un procedimiento que identifica el entorno o vecindad al punto de control, y a partir de allí genera una acción de control sólo para ese entorno (ajuste de parámetros). Siendo cada situación diferente a la actual una situación nueva, de tal manera que si el sistema se encuentra en un punto A de operación con parámetros de operación para el controlador P_A , y este cambia al punto de operación B , el controlador debe realizar la adaptación de sus parámetros a P_B . Si el sistema retorna a A el controlador debe realizar nuevamente la adaptación, esto debido a que el controlador adaptable no tiene memoria, lo que lo diferencia de un control inteligente que reconocería inmediatamente el punto A . Decimos entonces que un controlador adaptable no es inteligente por la ausencia de aprendizaje.

Para un sistema en lazo cerrado podemos definir las siguientes variables: \underline{R} como el vector de referencia del sistema, \underline{Y} el vector de salida del sistema, \underline{Z} el vector de las perturbaciones que afectan el sistema y \underline{P} que recoge los parámetros del controlador veamos el siguiente esquema:

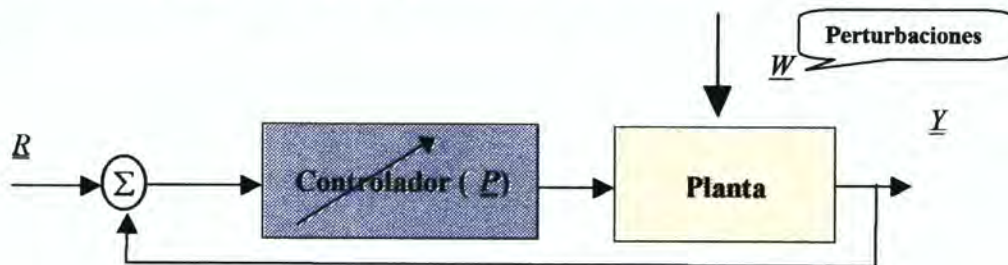


Figura 4-8 Esquema de un controlador adaptable

Entonces el comportamiento a adaptar seria $\underline{Y} = f(\underline{W}, \underline{R}, \underline{P})$.

En el caso de un PI clásico podríamos ajustar los parámetros del controlador (acción integral y proporcional) usando la lógica difusa de tal manera que se adapte a cambios en la planta. El siguiente esquema muestra el modelo que también es conocido como sistema híbrido ya que combina dos técnicas de control.

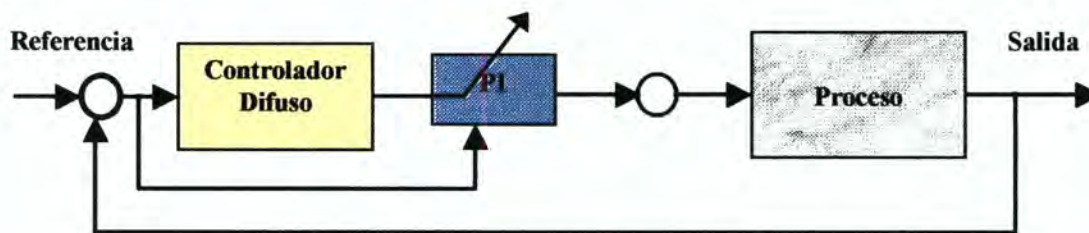


Figura 4-9 Adaptación difusa de un controlador PI

Control difuso adaptativo el primer controlador difuso adaptativo fue desarrollado en Queen Mary College por Procyk y Mamdani en 1979¹⁷, fue llamado SOC (the linguistic self-organizing controller) en el cual a través de un lazo de convergencia se mide el desempeño del controlador para modificar la base de reglas basándose en una convergencia métrica predecible. En la literatura se encuentran muchas aplicaciones de este método que se muestra en la figura 4-10. Una de las más recientes es la FMRLC “fuzzy model reference learning cotroller” que fue introducida por Kevin Passino y J. R. Layne, pero existen otras técnicas de control adaptativo.

¹⁷ PASSINO Kevin. M., Raul Ordoñez, Jon Zumberge, Jeffrey T. Spooner.:” Adaptive Fuzzy Control : Experiments and Comparative Analyses,” IEEE Transactions on Fuzzy Systems, Vol. 5, N 2, Mayo de 1997.

Los sistemas difusos adaptativos (adaptive fuzzy systems) son conocidos también como (self-organising). Surgen cuando se tiene dificultad en encontrar la base de reglas para una planta, o la necesidad de poder cambiar los parámetros de la base de reglas si la planta cambia.

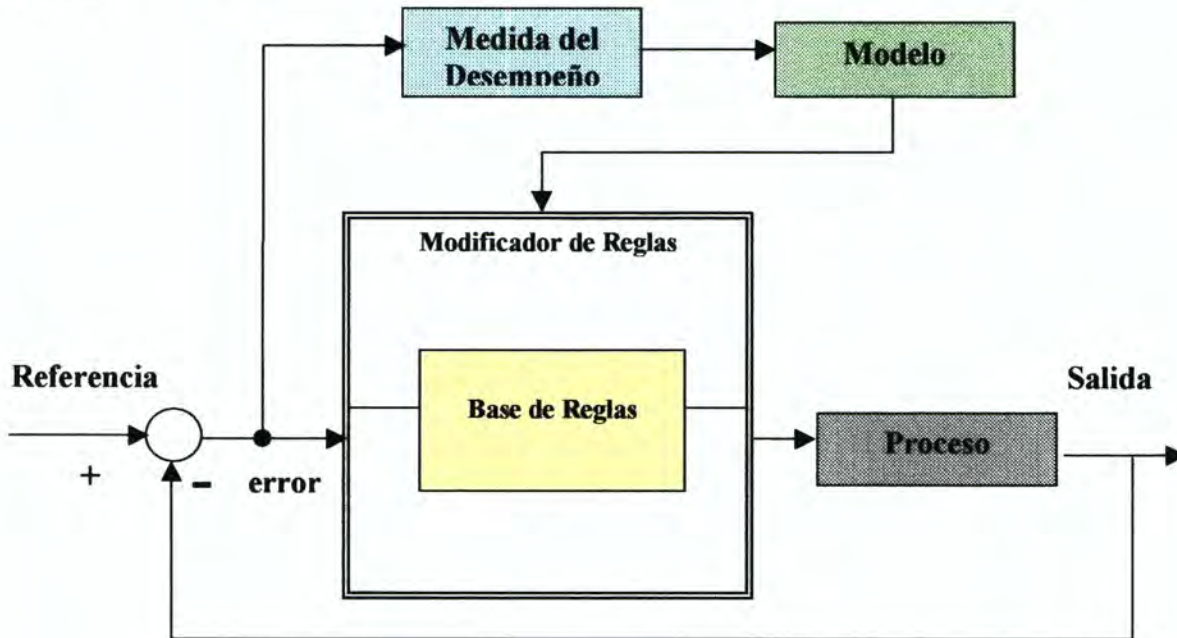


Figura 4-10 SOC de Procyk y Mamdani

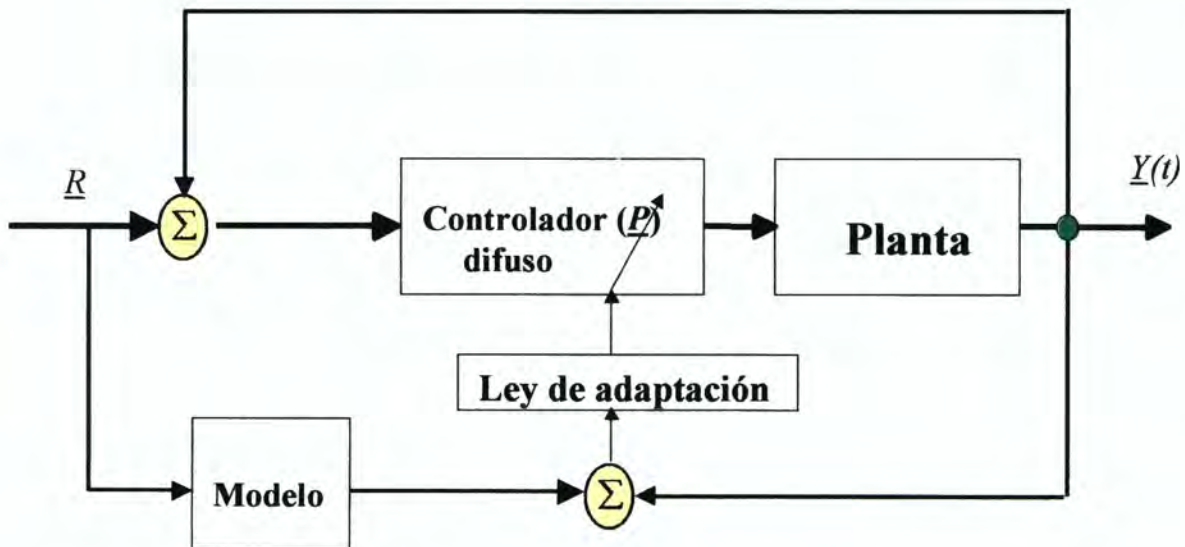


Figura 4-11 Estructura básica de un controlador difuso adaptativo

En la figura 4-11 tenemos el modelo básico de un controlador difuso adaptable que consta de:

El modelo de referencia se usa para especificar la respuesta ideal que el controlador difuso debe conseguir.

La ley de adaptación ajusta los parámetros del controlador difuso en línea de tal forma que la salida $y(t)$ siga la salida del modelo de referencia $y_m(t)$.

Ventajas del control difuso adaptativo sobre el control difuso no adaptativo:

- ☐ Mejor desempeño debido a que se puede autoajustar ante los cambios de la planta.
- ☐ Requiere menos información acerca de la planta ya que la ley de adaptación ayuda a conocer la dinámica de la planta durante su operación.

Desventajas:

- ☐ El sistema de control resultante es más difícil de analizar porque no sólo es no lineal, sino que también es variante en el tiempo.
- ☐ Su implementación es más costosa.

Los parámetros del controlador difuso que se pueden ajustar mediante la ley de control son:

- ☐ Los **factores de escala** para las variables de salida y de entrada.

📖 Elementos relacionales contenidos en la **matriz relacional**.

📖 Las **funciones de pertenencia** asociadas con las variables de entrada y de salida.

4.7. OTROS ENFOQUES

A pesar de los éxitos alcanzados por los controladores difusos, aún la industria es un poco escéptica en cuanto a su adopción. Esto se debe a que aún no se cuenta con la suficiente experiencia para crear de una manera práctica las reglas de control difusas. Otra razón es que se cuenta con muy pocos estudios sobre la estabilidad en lazo cerrado de los sistemas de control difuso.

Acorde con estas expectativas, muchos investigadores han propuesto estructuras híbridas de control las cuales combinan controladores del tipo PID con sistemas difusos. Los controladores PID son bastante simples de implementar y por esto son usados ampliamente en la práctica. Ellos pueden regular exitosamente la mayoría de los procesos industriales. Los estudios analíticos sobre la respuesta transitoria, estabilidad y disminución del error, garantizan el desempeño de los controladores PID sobre algunas condiciones de operación. Sin embargo, las capacidades de los controladores PID son significativamente reducidas cuando se aplican a sistemas con no linealidades tales como saturación, histéresis, zona muerta, etc.. En este caso los sistemas difusos pueden mejorar el desempeño de los controladores PID.

Actualmente se han propuesto estructuras mixtas de control que combinan el PID con los sistemas difusos. Estos controladores híbridos han surgido como una gran promesa y una tendencia muy justificable para el desarrollo y aplicación de los controladores difusos.

El principal aspecto es que **el controlador difuso no interactúa directamente con el proceso**, sino que **se encuentra situado en un alto nivel de planeación de la acción**. Los controladores difusos híbridos más difundidos son los que realizan el papel de supervisar el funcionamiento de algunos controladores locales, los cuales sí interactúan directamente con el proceso. Tal supervisión puede consistir en la conmutación entre los controladores locales o en la conmutación entre los conjuntos de parámetros dentro del mismo controlador.

Algunas de las arquitecturas híbridas más populares son:

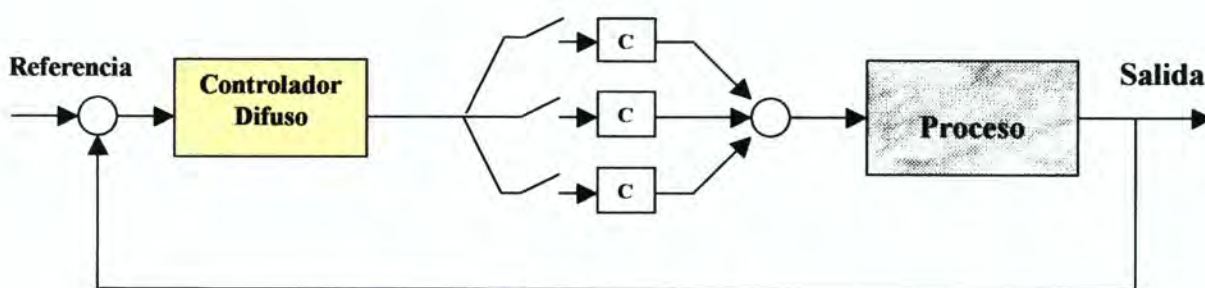


Figura 4-12 Arquitectura híbrida

En esta configuración, el controlador difuso se encarga de escoger el mejor compensador que de acuerdo al momento actual del sistema mejor satisfaga las necesidades del proceso.

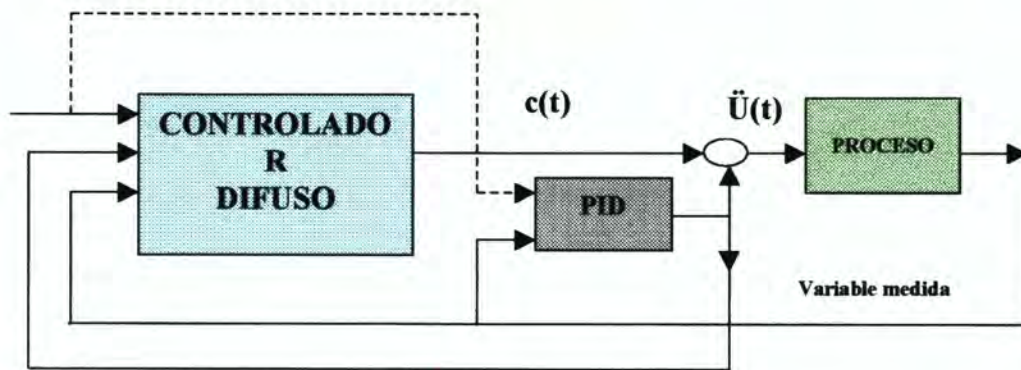


Figura 4-13 Controlador Híbrido (intervención del CD sobre el PID).

En la figura 4-13 se muestra otra configuración híbrida entre un controlador difuso y uno PID. Aquí, ambos controladores trabajan en paralelo. Aunque sus salidas son sumadas, la salida del controlador difuso es cero en condiciones normales de operación, dejándole al controlador PID el trabajo. Sólo cuando el CD detecte condiciones anormales en el proceso entonces entrará en acción. En este enfoque el CD compensa directamente la salida del PID. Esto permite un control con buenos resultados tanto en el estado transitorio como en el estacionario.

La señal de control preliminar, $u(t)$ es calculada por el PID basándose en el error de lazo cerrado $e(t) = Y_r - Y(t)$. Este valor de $u(t)$ es compensado por la salida $c(t)$ del CD. La señal final de control es calculada como $\ddot{u}(t) = u(t) + c(t)$.

Otra estructura híbrida conocida, es la del controlador PID precompensado. (Fig. 4-12) El conjunto de reglas difusas se construyen mediante ensayo y error con el fin de suprimir sobrepicos en la respuesta transitoria. Este esquema está comprobado que mejora el

desempeño y la robustez cuando controla la planta con no linealidades severas y desconocidas.

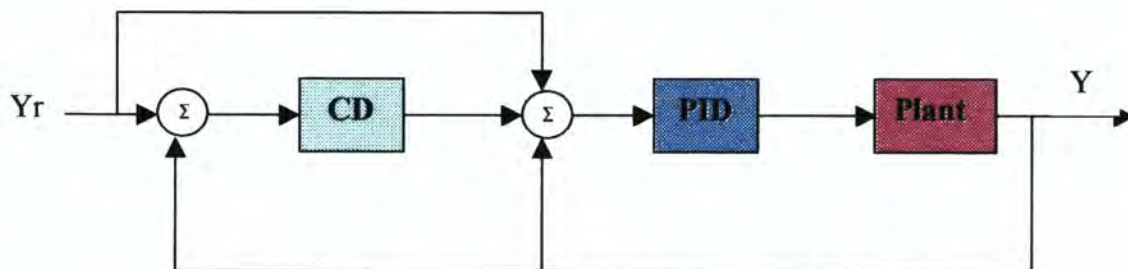


Figura 4-14 PID precompensado por un Controlador Difuso

La idea clave de la precompensación difusa es la de modificar el set-point con el fin de mejorar el desempeño del controlador PID cuando la planta presenta severas no linealidades.

Otro esquema híbrido permite que el CD modifique las ganancias del PID. En la Fig. se muestra cómo un CD "interpreta" las reacciones del proceso y sintoniza los parámetros P,I y D del controlador PID.

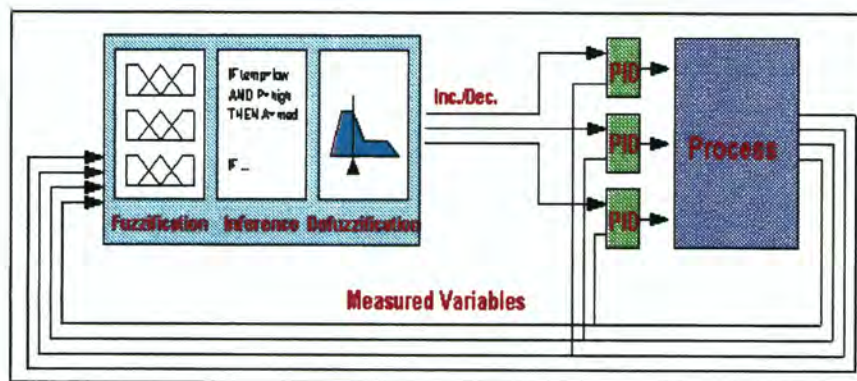


Figura 4-15 Modificación difusa de ganancias en PID

Un aspecto importante basado en la implementación de estas estructuras híbridas es que así como hacen parte de una estructura de control, no aparecen como competidores para las soluciones existentes. Esta técnica permite una utilización más eficiente de los controladores estándares existentes mediante la adición de un nivel de coordinación basado en un conocimiento adicional.

5. ESTABILIDAD Y ROBUSTEZ DE LOS CONTROLADORES DIFUSOS

En los últimos años, se han llevado a cabo muchos intentos para encontrar una teoría general sobre la estabilidad de los controladores difusos (CD). En algunos casos muy específicos, se ha logrado resolver el problema, pero en general, **se puede decir que aún no existe una teoría general sobre la estabilidad de los sistemas de control difuso (SCD).**

En algunos países, la discusión acerca de la lógica difusa y el análisis de estabilidad ha estado cerca de convertirse en una “guerra santa”. Esta discusión ha surgido por el hecho de la renuencia a aceptar que las técnicas de CD son **no analíticas** desde el punto de vista del análisis de la estabilidad.

Desde hace muchos años se viene estudiando el problema de la estabilidad y la robustez de los sistemas de control mediante las técnicas convencionales (analíticas), es por eso que se puede decir que existe un **culto a la analiticidad**. Este culto tiene mucho que ver con las críticas y la poca aceptación que tuvo hasta hace poco el CD (excepto en el Japón). Las críticas estaban centradas hacia la falta del análisis de estabilidad de los CD.

Como es bien sabido, existen sistemas para los cuales es muy difícil hallar una ley matemática que describa su dinámica, esto debido a que los procesos físicos que toman lugar allí son muy complicados. Cuando nos vemos enfrentados a tales sistemas, siempre

se invoca a la palabra clave: **no linealidad**. Los sistemas con no linealidades deben ser descritos mediante ecuaciones matemáticas no lineales, resultando así en una degradación de la analiticidad.

Para hacer un estudio de la estabilidad de un sistema, es preciso obtener un modelo matemático de éste, pero hay sistemas tan complejos, por ejemplo hornos de cemento (la primera aplicación industrial de CD) que no pueden ser modelados adecuadamente. Hoy en día existe gran cantidad de aplicaciones industriales con control difuso, lo cual ha hecho que cambie la visión que se tenía acerca de él. La carencia de una metodología para el análisis de estabilidad por parte del CD desde un punto de vista práctico no es una condición insalvable. La prueba de un prototipo es más importante que el análisis mismo de la estabilidad.

En el caso del tren subterráneo de la ciudad de Sendai (Japón), los investigadores realizaron alrededor de 300.000 simulaciones y actualmente se realizan unas 2.000 para probar el sistema, ya que no se puede “jugar” con un sistema subterráneo. Este tren subterráneo está operando desde el 15 de julio de 1987, convirtiéndose en un fuerte testimonio para la teoría. Es un sistema donde los principios de estabilidad y confiabilidad son de máxima importancia y se ha comprobado a través del tiempo su éxito.

La estabilidad como es entendida por los ingenieros de control, no es una propiedad del proceso que está siendo controlado sino una definición operacional expresada en el lenguaje matemático de la teoría analítica de control. Por ello, el CD se debe ver como una técnica que forma parte de la teoría no analítica de control. De todas maneras, la

estabilidad es una cuestión importante, pero existen diferentes maneras de llevar a cabo su estudio. Durante el análisis final de un sistema se debe contar con prototipos para lograr el propósito de su aprobación. Esta es una técnica ensayada y comprobada en la industria y no hay razón por la cual ella no sea eficaz también en los sistemas de control.

Debe notarse que no se ha comprobado que una base de reglas de control, la cual sea considerada estable dentro de su dominio difuso, donde la “ley de control” es lingüística por naturaleza o está dada en alguna forma numérica o funcional, satisfaga algún criterio analítico de funcionamiento. Se asume que el SCD es estable si las condiciones son satisfechas. Sin embargo, esta no es una prueba de que esta metodología será independiente de las condiciones iniciales.

Existen algunas aproximaciones a la teoría de estabilidad de controladores difusos como la técnica de la “Estabilidad Energética”, donde primero se define la energía de la salida del SCD. Para una entrada dada (X) y una relación difusa (R), se define la energía de salida del SCD como: $E(Y)=E(X*R)$ donde Y es la salida difusa, E es la energía asociada con la función de pertenencia Y y $(*)$ es el operador composicional de la regla de inferencia. Esta energía corresponde rigurosamente al área bajo la función de pertenencia. Para chequear la estabilidad de la relación, los cambios en la energía de salida, por ejemplo $\Delta E=E(Y_k)-E(Y_{k-1})$ están determinados, donde k y $k-1$ corresponden a instantes consecutivos de tiempo. Acorde con el teorema de Lyapunov, el sistema es estable si ΔE es negativa. Esto también indica que ΔE es independiente de las condiciones iniciales (X_0).

Otra técnica conocida es la de utilizar un controlador supervisorio para las condiciones de inestabilidad, aquí el CD es diseñado primero sin tener ninguna consideración sobre la estabilidad, luego otro controlador es adicionado al CD para cuidar de los requerimientos de estabilidad. La clave es cómo diseñar el controlador adicional para garantizar estabilidad. Debido a que lo que se requiere es que el CD realice la acción principal de control, el controlador adicional deberá ser un guardián en vez de un controlador principal. Por esto, se escoge al controlador adicional con las siguientes características: si el CD está trabajando bien, el controlador adicional permanecerá inactivo, si el sistema está tornándose inestable, el controlador adicional comienza su operación para garantizar la estabilidad. De esta manera el controlador adicional puede ser llamado un controlador supervisorio.

Un método propuesto por (SAEZ et al, 1996) para el diseño de un CD estable aprovecha los resultados de los métodos convencionales. Este método utiliza CD del tipo Takagi-Sugeno, en el que a cada consecuente se le asigna un controlador lineal. Si bien con este método es posible obtener un muy buen comportamiento del sistema realimentado, no se conoce un procedimiento para determinar de una manera sistemática, las premisas de las reglas que conforman el CD, más específicamente, las funciones de pertenencia de los conjuntos difusos asociados a las variables de entrada. Las condiciones de estabilidad para este método se basan en la aplicación del Primer Método de Lyapunov.

Otros estudios sobre la estabilidad de los SCD están basados en métodos estadísticos. Se propone el uso de ciertos índices de estabilidad, estos índices miden qué tan lejos/cerca está el sistema de perder la estabilidad (HOLGADO et al, 1994)

En cuanto a la robustez de los SCD se puede decir que tampoco existe una teoría bien elaborada y reconocida para realizar su análisis, lo que sí se puede decir es que la robustez de un SCD se encuentra implícita en las reglas, ya que éstas están basadas en el conocimiento que un experto humano tiene del sistema en cuestión. Cada regla representa un comportamiento local, es decir cada regla describe una reacción ante una situación. Si se encontraran problemas de estabilidad en esta situación, se puede modificar la regla respectiva, las demás reglas quedarían inalterables, dando paso a una optimización orientada hacia una meta. En un sistema de control convencional, alguna modificación hecha para mejorar determinada situación puede producir efectos indeseados en el sistema bajo otras situaciones

Por ahora la comprobación de la robustez de los CD se limita a las demostraciones físicas realizadas en numerosas conferencias internacionales. Un ejemplo de ello es la comparación hecha en la Universidad de Carolina del Sur entre un PID y un CD para el modelo de un carro¹⁸, en la cual se prueba que a pesar de variar en un amplio rango el peso del modelo del carro, no existe la necesidad de cambiar las reglas de control o los conjuntos difusos del CD, caso contrario del PID que requiere ser sintonizado de nuevo. De esta manera se demuestra experimentalmente la robustez de los CD.

¹⁸ PEREIRA John., "A comparison of PID y Fuzzy Control of a Model Car". En, IEEE Journal of Intelligents and Fuzzy Systems, Octubre de 1994, pp. 849-854.

6. APLICACIONES DEL CONTROL DIFUSO: CASOS

6.1. APLICACIÓN DE UN SISTEMA DE CONTROL HÍBRIDO PID-DIFUSO PARA UNA PLANTA PETROQUÍMICA.

Tomado de TANI Tetsuji, UTASHIRO Makoto, UMANO Motohide y TANAKA Kazuo., "Application of practical Fuzzy-PID Hybrid Control System To Petrochemical Plant" En Proceeding of Third IEEE International Conference on Fuzzy Systems. 1994 IEEE world congress on computational intelligence. Pp 1211-1216.

Los controladores PID son muy usados en las plantas petroquímicas, sin embargo no trabajan bien ante sistemas no lineales y variantes en el tiempo. En esta aplicación el controlador difuso compensa directamente la salida del PID. Esto permite el control de la planta con resultados bastante satisfactorios tanto en estado transitorio como en el estacionario.

Descripción del proceso: En la Figura6-1 se muestra el proceso de la producción de nafta hidro-desulfurada, en la cual se desulfura el combustible entrante, que contiene un ácido de gas de petróleo licuado (ácido GPL) para producir gasolina y GPL en las siguientes etapas:

1. El combustible es mezclado con hidrógeno gaseoso y son bombeados hacia un precalentador para lograr la temperatura de reacción. Esta mezcla luego es enviada a un reactor para su desulfuración.

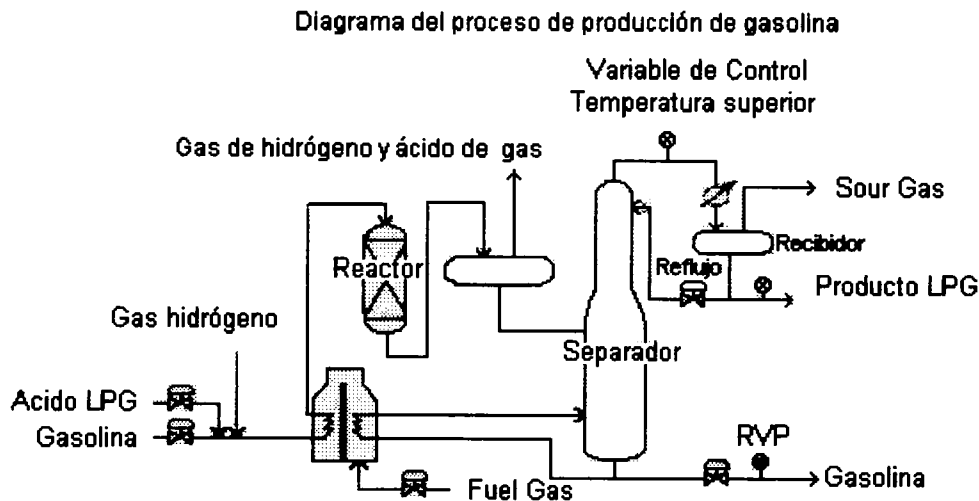


Figura 6-1 Proceso de producción

2. La mezcla a la salida del reactor es separada en el ácido de gas y en combustible (nafta y GPL) dentro de un condensador.
3. El combustible (nafta y GPL) es enviado al separador en donde se divide en gasolina y en GPL, este último es extraído de la parte superior del separador y la gasolina de la parte inferior.

Problemas de Control: La planta presenta diversas características que afectan fuertemente la calidad del producto, como por ejemplo:

- (i) Cuando la temperatura en el calentador es muy alta, la calidad del GPL es muy pobre y cuando la temperatura es muy baja, la calidad de la gasolina decrece ya que la RVP (Reid Vapor Pressure) sube. Es por esto que la temperatura del calentador para la mezcla debe ser controlada en un punto ideal.

- (ii) Como toma mucho tiempo medir la RVP de la gasolina, es difícil de determinar la temperatura del calentador.

Estructura del Sistema: El sistema de control híbrido PID-Difuso permite controlar la temperatura superior del separador reemplazando en esta función a un operador experto. El sistema consta de dos secciones. Una sección PID y una sección de razonamiento difuso. (ver figura 6-2)

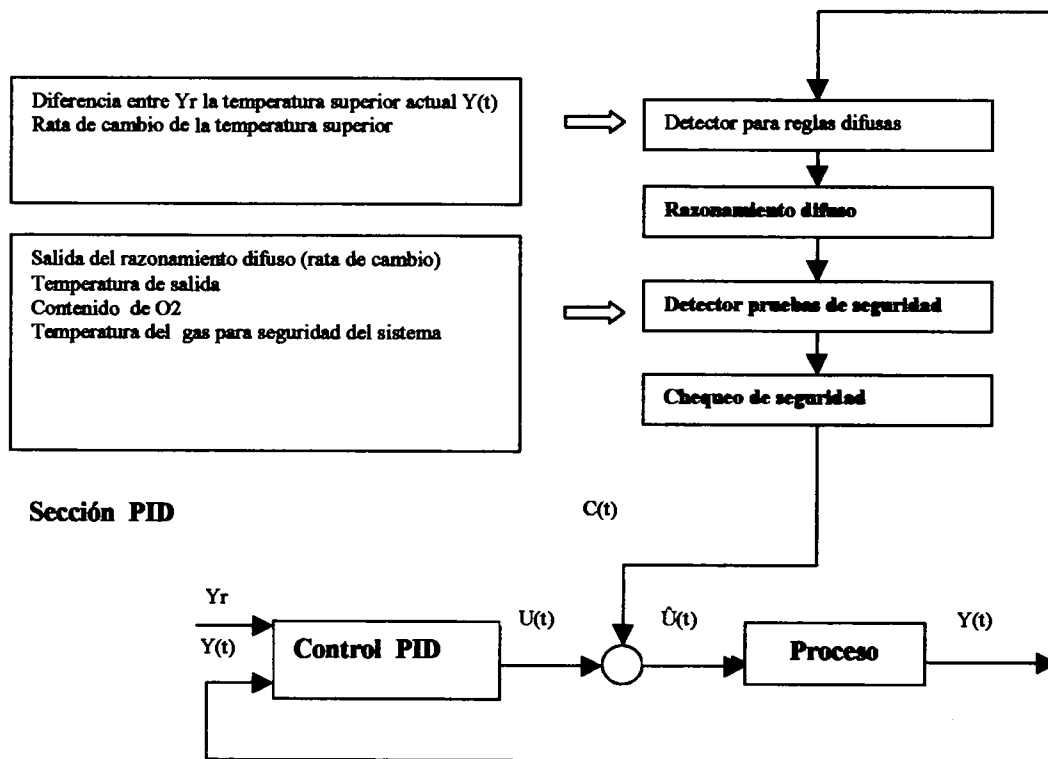


Figura 6-2 Configuración del control híbrido fuzzy-PID

Un valor preliminar de la temperatura superior $u(t)$ es calculado por el PID basándose en el error de lazo cerrado $e(t)=Y_r-Y(t)$, donde Y_r es el valor deseado para la temperatura superior y $Y(t)$ es la variable del proceso. El PID tiene la forma estándar:

$$u(t) = K_c \left[e(t) + T_d \frac{de(t)}{dt} + \frac{1}{T_i} \int e(t) dt \right]$$

Donde K_c , K_d y K_i son las constantes de ganancia proporcional, derivativa e integral respectivamente.

El valor preliminar de salida $u(t)$ es compensado por la salida $c(t)$ de la sección de razonamiento difuso, en el cual sus salidas corresponden a singletons. Las reglas difusas son de la forma:

Ri: SI X_1 es A_{1i} y X_n es A_{ni} ENTONCES $Y_i = b_i$,

Donde X_j es la j -ésima variable de entrada para las i -ésimas reglas. Y_i es la variable de salida para la i -ésima regla. A_{ij} es un conjunto difuso y b_i es un valor singleton.

Dada una entrada (x_1, x_2, \dots, x_n) en el tiempo t , el grado de matching en la premisa para la i -ésima regla es calculado así:

$$W_i = A_{1i}(x_1) * A_{2i}(x_2) * \dots * A_{ni}(x_n)$$

Sección de razonamiento difuso: La temperatura superior es controlada por la salida compensada del PID, la cual es calculada así:

$$\hat{U}(t) = U(t) + C(t)$$

Resultados: Este sistema de control se aplicó a una planta de producción de nafta hidro-desulfurada en la refinería de Idemitsu en Hokkaido.

En estado estacionario (Figura 6-3), el resultado de usar el controlador propuesto muestra que la temperatura superior es controlada con una desviación de 1.5 C.

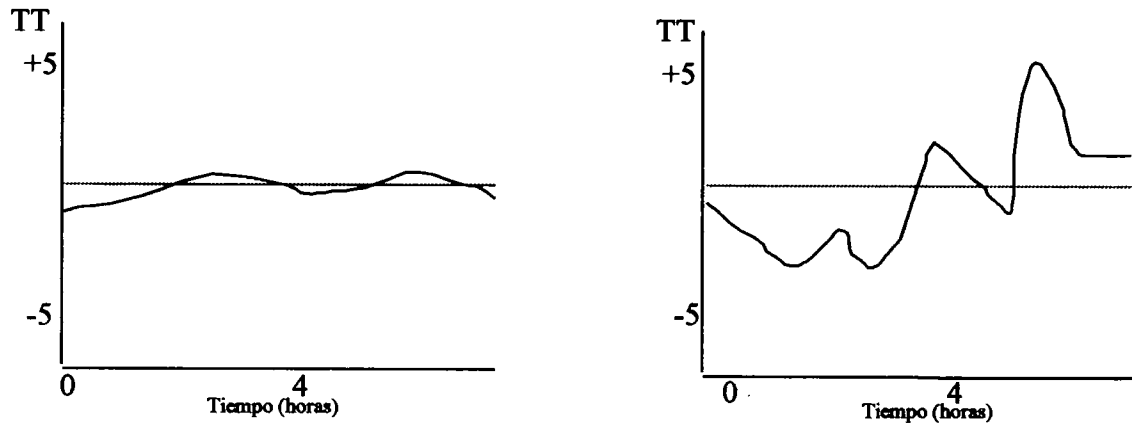


Figura 6-3 Operación con el control híbrido PID-Difuso

Inversamente, con el controlador PID solo, el control podría no ser estable y presentar una variación de ± 10 C en el mejor de los casos.

El sistema es muy efectivo en el control de la planta actual, no solo en el estado estacionario, sino también durante el transitorio.

6.2. CONTROL DE UNA PLANTA INCINERADORA DE BASURAS MEDIANTE TECNICAS DIFUSAS

Tomado de Von ALTROCK Constantin, KRAUSE Bernhard, "Development of a Fuzzy Knowledge-Based System for the Control of a Refuse Incineration Plant" En fuzzyTECH Application Paper (http://www.fuzzytech.com/e_a_mull.htm)

Una planta para incinerar basuras involucra un proceso complejo cuyos problemas de control presentan muchas variables, los cuales no se pueden resolver convencionalmente al hallar un modelo matemático exacto del proceso. La utilidad de las avanzadas técnicas difusas es demostrada por un controlador implementado para una planta de incineración de basuras en la ciudad alemana de Hamburgo.

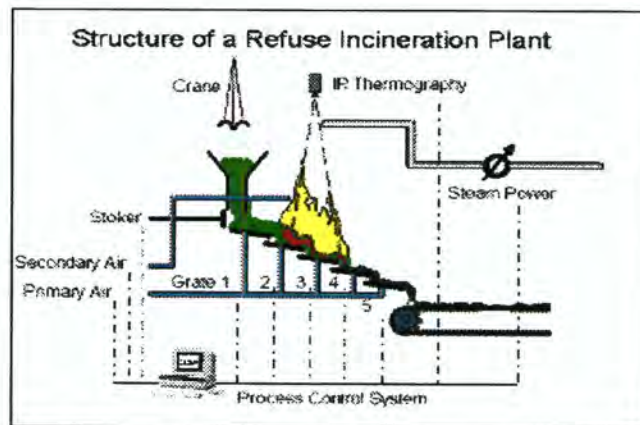


Figura 6-4 Estructura de la planta incineradora de basuras

Debido a la heterogénea composición de las basuras, muchas plantas de incineración eran controladas solo en forma parcial mediante la observación de la cámara de combustión por un operario. (ver figura 6-4)

Durante la combustión el sistema de control debe mantener las siguientes condiciones:

- ❑ Control de la concentración de O_2 en el gas que sale de la chimenea para mantenerlo sobre valores constantes.
- ❑ Mantener a la salida una uniformidad térmica.

- ☐ Mantener unas condiciones para un flujo óptimo entre la basura y el primer quemador y así prevenir problemas con emisiones indeseables.

El método utilizado en la automatización de este proceso involucra el registro de la combustión por termografía infrarroja. (ver figura 6-5)

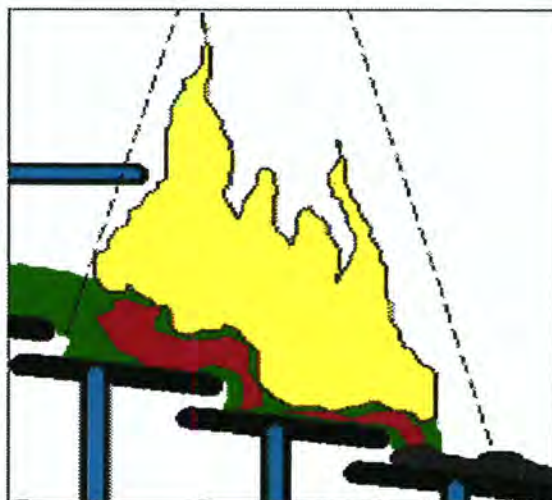


Figura 6-5 Área de observación cubierta por la cámara infrarroja

La estructura del sistema de control se divide en tres partes: la primera corresponde al circuito de control para la capacidad de vaporización, la segunda controla el movimiento de las basuras dentro de la cámara de combustión y la tercera se encarga de optimizar la combustión. (ver figura 6-6)

La utilización de la lógica difusa permite la integración de muchos parámetros disímiles como por ejemplo la capacidad de vaporización además de varios parámetros para la identificación de la posición del fuego y su altura mediante técnicas de termografía infrarroja.

Esta capacidad de optimización del fuego asegura un patrón de combustión aún mucho mayor.

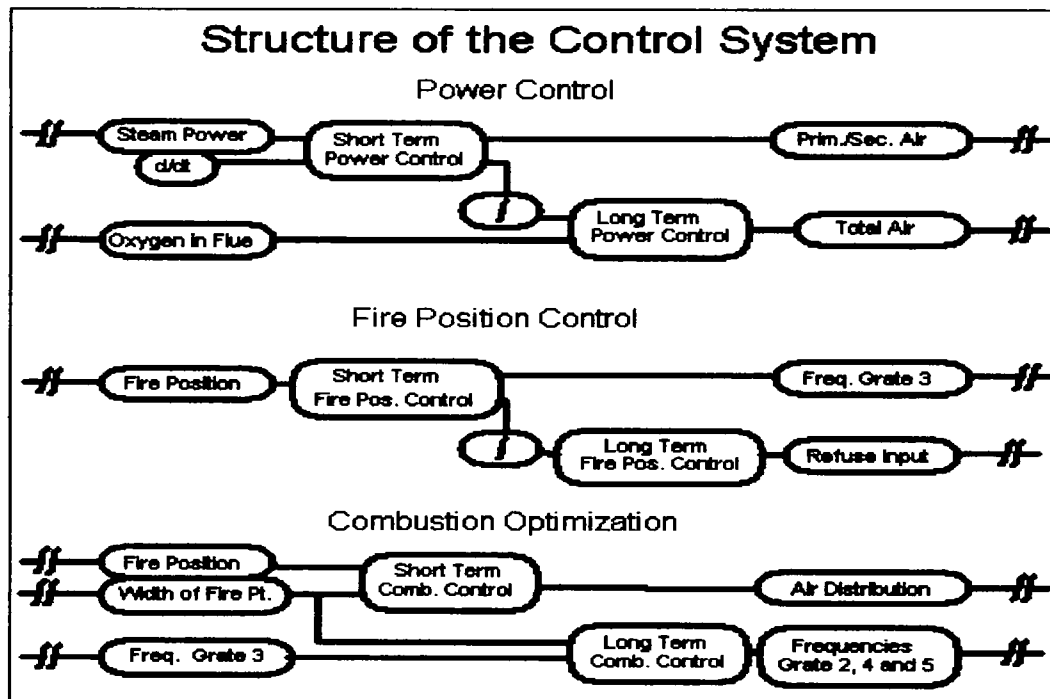


Figura 6-6 Estructura del sistema de control difuso

Como resultado de ello es posible reducir el contenido de monóxido de carbono en el gas emitido hacia la atmósfera.

6.3. LIMITADOR DE VELOCIDAD PARA CAMIONES DE CARGA

Tomado de Von ALTROCK Constantin, KRAUSE Bernhard, NIELD Eric, Mike LORD, y Alexander BAKER "Truck Speed Limiter Control by Fuzzy Logic" En : Fuzzy Logic '94 Conference held, San Francisco., Septiembre de 1994.

URL: www.fuzzytech.com/e_a_spe.htm

En Europa, los camiones cuya carga exceda las 12 toneladas, deben estar equipadas con un controlador que limite su velocidad máxima hasta 86 Km./h.

Este limitador trabaja del siguiente modo: cuando el camión se aproxime a la velocidad máxima, la válvula neumática reduce el obturador de la bomba de combustible para que la velocidad máxima no sea sobrepasada.

Si el conductor presiona el acelerador aún más, el limitador de velocidad debe asegurar el desplazamiento del camión a la velocidad máxima. (ver figura 6-7)

Debido al tiempo muerto y a no linealidades involucradas con esta acción de control, ocurre un sobre paso y oscilaciones al usar controladores convencionales.

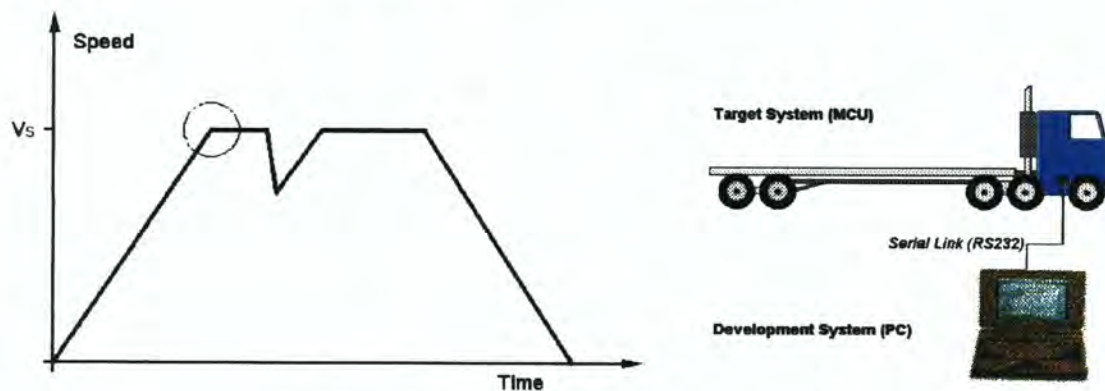


Figura 6-7 Curva de velocidad vs tiempo.

El popular PID trabaja bien con procesos que tienen un punto de operación fijo, pero para el limitador de velocidad del camión el punto de operación se mueve debido a las diferentes situaciones: como si el camión va subiendo o bajando, o si viaja vacío o con el cupo completo y además las características de la válvula neumática y de la inyección de

combustible son grandemente no lineales y también varían de un camión a otro. (ver figura 6-8)

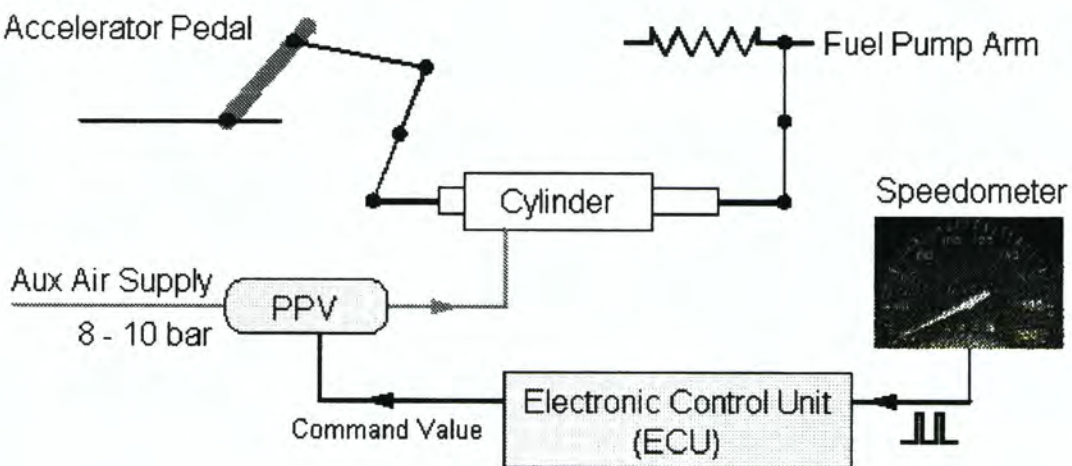


Figura 6-8 Diagrama del sistema.

La unidad electrónica de control compara la señal digital proveniente del velocímetro con la máxima velocidad preestablecida, basada en esto calcula el valor para la válvula de pulsos proporcionales que controla la presión del aire en el cilindro. (ver figura 6-9)

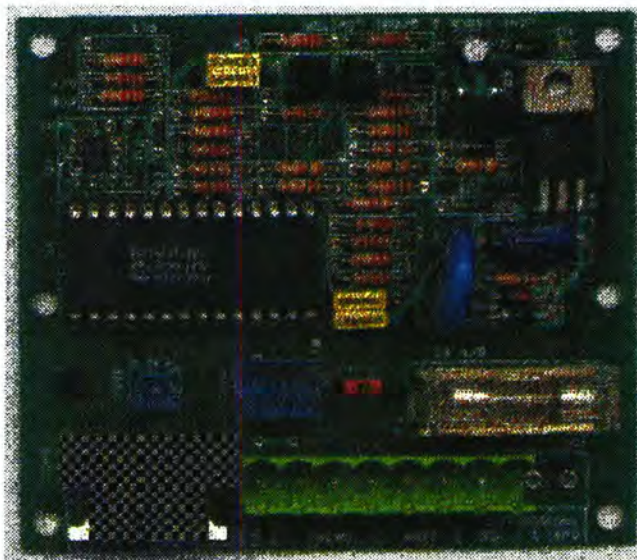


Figura 6-9 Unidad electrónica de control

cada valor en un conjunto difuso definido en U ; G es una regla sintáctica para generar los nombres de los valores de x ; y M es una regla semántica para asociar cada valor con su significado. Veamos un ejemplo:

Interpretando la velocidad como una variable lingüística tendríamos $x = \text{velocidad}$

$$T(x) = T(\text{velocidad}) = \{\text{muy lenta, lenta, media, más o menos rápida, rápida}\}$$

$U = [0, \text{velocidad máxima}]$ si se toma la *velocidad máxima* como 100 Kph

$U = [0, 100]$, cada término en $T(x)$ estará caracterizado por un conjunto difuso en el universo de discurso $U=[0, 100]$, la regla G estaría implícita en el hecho de definir por medio de nuestra heurística y nuestro lenguaje los conjuntos difusos etiquetados con los términos {muy lenta, lenta, media, más o menos rápida, rápida}, igual se puede definir tan solo tres conjuntos para la misma variable velocidad {lenta, media, alta}. La regla M la asociaríamos a interpretar "muy lenta" como "una *velocidad* por debajo o cercana a 20 Kph," "lenta" como "una *velocidad* cercana a 35 Kph," "media" como "una *velocidad* cercana a 50 kph," "más o menos rápida" como "una *velocidad* alrededor de 65 Kph," "rápida" como "una *velocidad* cercana o mayor a 80 Kph". La representación de la variable lingüística *velocidad*, donde $\mu(\text{velocidad})$ representa los valores de la función de pertenencia de los cinco conjuntos difusos se muestra en la figura 3-4.

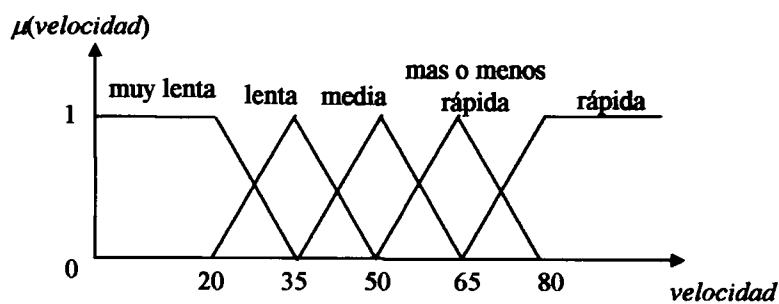


Figura 3-4 Representación de la variable lingüística *velocidad*

Definición intuitiva¹¹, *Variable lingüística*: Si una variable puede tomar palabras en lenguaje natural (por ejemplo, pequeño, rápido, etc.) como sus valores, esta variable se puede definir como una variable lingüística. Estas palabras son usualmente etiquetas de conjuntos difusos.

Una variable puede tomar palabras o números como sus valores, como se vio en el ejemplo anterior, la variable lingüística *velocidad* puede tomar "muy lenta," "lenta," "media," "más o menos rápido," o "rápido" como sus valores, pero también puede tomar valores de números reales en el intervalo $[0, 100]$.

3.3 REGLAS IF-THEN DIFUSAS

Las reglas IF-THEN difusas o reglas fuzzy son proposiciones simples o compuestas. Ellas resultan de una conjunción entre proposiciones simples difusas. Estas reglas están compuestas por una premisa y una conclusión. Su estructura clásica es la siguiente:

$$\text{If } \langle \text{premisa} \rangle \text{ then } \langle \text{conclusión} \rangle \quad (3.7)$$

Cuando la premisa es una proposición simple difusa, la regla se describe de la siguiente forma: **If** $\langle x \text{ es } A \rangle$ **then** $\langle \text{conclusión} \rangle$. La x es una variable lingüística, definida en U , que es llamado *el universo de discurso*. A es un termino lingüístico tomado de $T(x)$, que

¹¹ WANG, Op. Cit., p. 43

Los casos de premisas compuestas no cambian la estructura general de una regla. Varias proposiciones simples difusas se hacen simplemente conjugando por medio de un operador conjunción: *AND*. En el caso de que dos variables difusas estén incluidas en la premisa, la regla toma la siguiente forma:

$$\text{If } \langle x \text{ es } A \text{ and } y \text{ es } B \rangle \text{ then } \langle \text{conclusión} \rangle \quad (3.8)$$

La construcción de un conjunto de reglas es un paso importante para la estabilidad del modelo de un proceso o construcción de un controlador difuso. Si este conjunto es conocido, la conclusión debería ser ahora evaluada cuando la premisa no es exactamente $\langle x \text{ es } A \rangle$ pero $\langle x \text{ es } A \rangle$. Esto se hace usando un modelo de razonamiento aproximado (la regla del generalizado modus ponens extensión del clásico modus ponens). El mecanismo de raciocinio es el siguiente:

Regla: **if** $\langle x \text{ es } A \rangle$ **then** $\langle y \text{ es } B \rangle$

AND

observación: $\langle x \text{ es } A \rangle$

THEN

Conclusión: $\langle y \text{ es } B \rangle$

La función de pertenencia $f_B(y)$ se obtiene combinando la regla con la observación. Esta combinación es hecha por un operador *T-norma*. La evaluación de $f_B(y)$ para todos los y es obtenida por la relación

$$f_B(y) = \sup_{x \in X} \{ T(f_A(x), r(x,y)) \} \quad (3.9)$$

Donde $f_A(x)$ está reducido a un singleton difuso normalizado, la relación previa transforma $f_B(x) = r(x_0, y)$. En ambas expresiones, $r(x,y)$ es una relación difusa trasladando la implicación entre las dos proposiciones difusas, $\langle x \text{ es } A \rangle$ y $\langle y \text{ es } B \rangle$. Esta relación esta definida por $r(x,y) = \text{imp}(f_A(x), f_B(y))$ para una *imp* función desde $[0,1] \times [0,1]$ en $[0,1]$.

Considerando un caso donde el operador conjunción es el mínimo y la función de implicación es también el mínimo. Nosotros suponemos que $f_A(x)$ es reducida a un normalizado singleton difuso. En este caso, la evaluación de $f_B(y)$ para todos y de Y es obtenida por

$$f_B(y) = \min (f_A(x_0), f_B(y)) \quad (3.10)$$

Este principio corresponde al modelo de Mamdani. La evaluación de la función de pertenencia f_B da como resultado una proyección del grado de pertenencia de observación x_0 a través del término lingüístico A para todos los puntos y de Y .

Existen dos formas principales de obtener las reglas difusa IF-THEN:

1. Preguntar a expertos humanos.
2. Usar algoritmos de entrenamiento basados en medición de datos.

El primero es la mejor forma de obtener las reglas, pero los expertos humanos en muchos casos no proveen un suficiente número de reglas.

Si las reglas son provistas por un experto humano, entonces las funciones de pertenencia deberían ser especificadas por el experto debido a que estas funciones son parte integral del conocimiento del experto.

Una definición formal del reglas difusa IF-THEN¹²:

$$R^{(l)} : \text{IF } x_1 \text{ es } F_1^l \text{ y } \dots \text{ y } x_n \text{ es } F_n^l, \text{ THEN } y \text{ es } G^l \quad (3.11)$$

donde F_i^l y G^l son conjuntos difusos, $\underline{x} = (x_1, \dots, x_n)^T \in U$ y $y \in V$ son variables lingüísticas de entradas y salidas respectivamente, y $l = 1, 2, \dots, M$. Cada regla define un conjunto difuso $F_1^l \times \dots \times F_n^l \rightarrow G^l$ en el producto espacial $U \times V$.

3.4 FUZZIFICACIÓN

La fuzzificación realiza un mapeo desde un punto crisp $\underline{x} = (x_1, \dots, x_n)^T \in U$ a un conjunto difuso A en U . Hay (por lo menos) dos opciones posibles de este mapeo.

¹² WANG, Op. Cit., p.51

▣ **Fuzzificación Singleton:** A es un singleton difuso para los x , tales que para un punto $p \in U$, su pertenencia se defina como $\mu_A(p) = 1$ y $\mu_A(x) = 0$ para todos los otros $x \in U$ con $x \neq p$. Otra definición, si el soporte del conjunto difuso A es un simple punto en U para el cual su pertenencia $\mu_A = 1$, A es un singleton difuso.

▣ **Fuzzificación Nosingleton:** A es un nosingleton difuso si la función de pertenencia que lo caracteriza es de forma triangular, gaussiana, trapezoidal, etc.

El singleton difuso tiende a ser usado en muchas aplicaciones, aunque el nosingleton puede ser usada cuando las entradas son afectadas por ruido.

La fuzzificación esta influenciada por la partición que se haga del universo de discurso U y las formas que se adopten para las funciones de pertenencia μ_x que caracterizan los conjuntos difusos en U .

3.5 BASE DE REGLAS Y MOTOR DE INFERENCIA

Base de reglas una base de reglas difusas consta de una colección de reglas difusas IF-THEN en la siguiente forma:

$$R^{(j)} : \text{IF } x_i \text{ es } F_i^j \text{ y } \dots \text{ y } x_n \text{ es } F_n^j, \text{ THEN } y \text{ es } G^j \quad (3.12)$$

donde F_i^l y G^l son conjuntos difusos, $\underline{x} = (x_1, \dots, x_n)^T \in U$ y $y \in V$ son variables lingüísticas de entradas y salida del sistema difuso respectivamente, y $l = 1, 2, \dots, M$. Cada regla define un conjunto difuso $F_i^l \times \dots \times F_n^l \rightarrow G^l$ en el producto espacial $U \times V$.

Sin pérdida de generalidad, consideramos multiple-entrada-simple-salida para el sistema difuso, ya que un sistema de múltiple salida se puede siempre descomponer en un grupo de sistemas de simple-salida. M es el número de reglas difusas y n el número de variables de entrada.

La base de reglas es el corazón del sistema de lógica difusa en el sentido que todos los otros tres componentes son usados para interpretar estas reglas y hacerlas útiles para problemas específicos. La práctica muestra que las reglas difusas IF-THEN en la forma de (3.12) proveen una estructura muy conveniente para que expertos humanos puedan expresar su dominio del conocimiento.

Motor de inferencia en un motor de inferencia difusa, los principios de lógica difusas son usados para combinar las reglas difusas IF-THEN en la base de reglas en un mapeo desde conjuntos difusos en $U = U_1 \times \dots \times U_n$ a conjuntos difusos en V .

Debemos definir cómo interpretar una regla difusa IF-THEN de la forma (3.12).

Interpretación de una regla difusa una regla difusa IF-THEN es interpretada como una implicación difusa $F_1^I \times \dots \times F_n^I \rightarrow G^I$ en $U \times V$. Sea un conjunto difuso A' en U la entrada a un motor de inferencia; entonces cada regla difusa IF-THEN determina un conjunto difuso B' en V usando la composición sup-star. Que es,

$$\mu_{B'}(y) = \sup_{x \in U} [\mu_{F_1^I \times \dots \times F_n^I}(x, y) * \mu_{A'}(x)] \quad (3.13)$$

Veamos las interpretaciones de las reglas IF-THEN más comúnmente usadas. Por simplicidad $F_1^I \times \dots \times F_n^I = A$, $G^I = B$ y $A \rightarrow B$ denota la regla.

☐ Regla operador mínimo de implicación difusas:

$$A \rightarrow B(x, y) = \min\{ \mu_A(x), \mu_B(y) \} \quad (3.14)$$

☐ Regla operador producto de implicación difusas:

$$A \rightarrow B(x, y) = \mu_A(x) \mu_B(y) \quad (3.15)$$

☐ Regla aritmética de implicación difusas:

$$A \rightarrow B(x, y) = \min\{ 1, 1 - \mu_A(x) + \mu_B(y) \} \quad (3.16)$$

□ Regla max-min de implicación difusas:

$$A \rightarrow B(\underline{x}, y) = \max\{\min[\mu_A(\underline{x}), \mu_B(y)], 1 - \mu_A(\underline{x})\} \quad (3.17)$$

□ Regla booleana de implicación difusas:

$$A \rightarrow B(\underline{x}, y) = \max\{1 - \mu_A(\underline{x}), \mu_B(y)\} \quad (3.18)$$

□ Regla Goguen's de implicación difusa:

$$A \rightarrow B(\underline{x}, y) = 1 ; \mu_A(\underline{x}) \leq \mu_B(y) \quad (3.19)$$

$$A \rightarrow B(\underline{x}, y) = \mu_B(y) / \mu_A(\underline{x}) ; \mu_A(\underline{x}) > \mu_B(y)$$

En (3.14) y (3.19) $\mu_A(\underline{x}) = \mu_{F_1^I \times \dots \times F_n^I}(\underline{x})$ y se define con la operación mínimo

$$\min\{\mu_{F_1^I \times \dots \times F_n^I}(\underline{x})\}$$

3.6 DEFUZZIFICACIÓN

La defuzzificación ejecuta un mapeo desde un conjunto difuso V a un punto crisp $y \in V$.

Hay (por lo menos) tres opciones posibles de este mapeo.

▣ **Defuzzificación Máximo**, definido como

$$y = \operatorname{argsup}_{y \in V} (\mu_B(y)) \quad (3.20)$$

donde $\mu_B(y)$ está dada por la ecuación $\mu_B(y) = \mu_{B^1}(y) + \dots + \mu_{B^M}(y)$

▣ **Defuzzificación del centro de gravedad**, definido como

$$y(\underline{x}) = \frac{\sum_{l=1}^M y^{-l} (\mu_{B^l}(y^{-l}))}{\sum_{l=1}^M (\mu_{B^l}(y^{-l}))} \quad (3.21)$$

donde y^{-l} es el centro del conjunto difuso G^l , que es, el punto en V en el que $\mu_{G^l}(y)$ alcanza su máximo valor, y $\mu_{G^l}(y)$ está dado por

$$\mu_{B^l}(y) = \operatorname{Sup}_{\underline{x} \in U} \left[\mu_{F_1^l \times \dots \times F_n^l \rightarrow G^l}(x, y) * \mu_{A^l}(\underline{x}) \right] \quad (3.22)$$

El conjunto difuso B^l es la union de M conjuntos difusos B^l con función de pertenencia

$$\mu_{B^l}(y) = \mu_{B^1}(y) \oplus \dots \oplus \mu_{B^M}(y)$$

☐ **Defuzzificación del centro de gravedad modificado**, definido como

$$y(\underline{x}) = \frac{\sum_{l=1}^M y^{-l} (\mu_{B^l}(y^{-l}) / \delta^l)}{\sum_{l=1}^M (\mu_{B^l}(y^{-l}) / \delta^l)} \quad (3.23)$$

donde δ^l es un parámetro que caracteriza la forma de $\mu_{G^l}(y)$.

3.7 APLICACIONES DE LÓGICA DIFUSA

El hecho más importante para el éxito de la lógica difusa ha sido la amplitud de sus aplicaciones. Por ejemplo hoy en día se encuentran productos para el control de procesos industriales, para el sector automotriz, en aparatos de uso doméstico, bioingeniería y en aplicaciones financieras. En la tabla 3-1 se ilustra este punto en términos de una lista parcial de aplicaciones de lógica difusa en el ambiente industrial. En cada una de estas aplicaciones a pesar de las obvias diferencias en alcance y/o manera de implementación, la lógica difusa juega un papel central en el modelado de una apropiada estrategia de control lingüística o basada en reglas. Otro factor que ha contribuido al desarrollo de aplicaciones de lógica difusa es la combinación de estrategias de control tradicionales (control clásico, moderno y inteligente) con lógica difusa conocidos como híbridos. Además, en la mayoría de las aplicaciones listadas en la tabla 3-1, La lógica difusa sirve de puente entre el vacío entre el procesamiento simbólico y la computación numérica, como consecuencia de esto se ha incrementado su uso en aplicaciones en ingeniería de control.

Tabla 3-1 Aplicaciones de lógica difusa en control

| Productos de Consumo | Automóviles y Generación de energía | Control de procesos industriales | Robótica e Industria |
|---|--|---|--|
| <ul style="list-style-type: none"> • Cámaras y Videocámaras (Canon, Minolta, Ricoh, Sanyo) • Máquinas lavadoras (AEG, Sharp, Goldstar) • Refrigeradores (Whirlpool) • Máquinas de vacío (Phillips, Siemens) | <ul style="list-style-type: none"> • Potencia en trenes y control de transmisión (GM, Saturn, Honda, Mazda) • Ingeniería de control (Nissan) | <ul style="list-style-type: none"> • Hornos de cemento, plantas incineradoras (K. L. Smith, Denmark) • Refinería, destilación y otros procesos químicos | <ul style="list-style-type: none"> • Máquina de descarga eléctrica (Mitsubishi) |

4. CONTROL DIFUSO

Los sistemas de control convencional han requerido de una descripción cualitativa y precisa de la relación entrada/salida usando complejas ecuaciones matemáticas (modelo matemático). Sin embargo, el control difuso acepta descripciones de sistemas de control que contienen un lenguaje vago o impreciso a pesar de usar lógica o ecuaciones matemáticas, y estos parámetros son formados como “conocimiento”.

Para implementar dicha estrategia de control es necesario encontrar una forma de convertir la descripción lingüística en valores numéricos para que los datos de entrada puedan generar los datos de salida. Por este motivo y como se vio en el capítulo 2 el tipo de sistemas difusos más apropiado para la mayoría de aplicaciones en control es el de Mandani (o sistema difuso con fuzzificador y defuzzificador) ya que éste provee las interfaces de fuzzificación y defuzzificación que permiten convertir datos numéricos reales provenientes de sensores en conjuntos difusos que a través del motor de inferencia y la base de reglas, permiten generar la acción de control apropiada para llevar el sistema a las condiciones deseadas dando solución al problema de control.

En este trabajo tomaremos el modelo de Mandani como referencia para las aplicaciones de control por ser el más intuitivo, pero sin descartar el modelo Takagi-Sugeno que para

ciertas aplicaciones puede ser una mejor alternativa gracias a que es más eficiente computacionalmente.

4.1. DESARROLLO HISTÓRICO

A partir de la primera publicación de Lotfi Zadeh en 1965 que introdujo el concepto de conjuntos difusos, y después de años de debates académicos sobre sus méritos, la lógica difusa emergió como una alternativa a la lógica clásica binaria en aplicaciones que van desde control de procesos industriales a productos de consumo para el hogar, el aeroespacio y la bioingeniería, constituyéndose en una herramienta para el control en situaciones donde el control clásico o moderno no ha dado una solución apropiada al problema de control.

El primer sistema de control implementado usando conjuntos difusos fue el propuesto en Europa por Ebrahim Mandani en 1975 para controlar una máquina de vapor y una caldera que usaba una síntesis de un conjunto de reglas de control obtenidas de un operador experto. Mandani se basó en una publicación de Lotfi Zadeh de 1973 "*Outline of a new approach to the analysis of complex systems and decision processes*". Después de esta aplicación surgieron diferentes grupos de investigadores que basados en la teoría de conjunto difusos y la lógica difusa propuesta por Zadeh implementaron aplicaciones destacándose un gran desarrollo de estas en el Japón. Estos desarrollos se encuentran resumidos en la tabla No 4.1 de la cual resaltamos la aplicación propuesta por Takagi y Sugeno en 1985 y el chip difuso propuesto por Togai y Watanabe.

Tabla 4.1 Desarrollos Históricos de Control Difuso

| Año | Autor | Contribución |
|------------|---------------------------|---|
| 1972 | Zadeh | Una razón para el control difuso |
| 1973 | Zadeh | Enfoque lingüístico |
| 1974 | Mamdani & Assilian | Control de una máquina de vapor |
| 1976 | Brae & Rutherford | Análisis de algoritmos de control |
| 1977 | Ostergaard | Control de un intercambiador de calor y un horno de cemento |
| 1977 | Willacyss et al. | Control difuso óptimo |
| 1979 | Kolomov et al. | Autómata finito |
| 1980 | Hitachi | ATO operación automática de un tren |
| 1980 | Tong et al. | Control del proceso de aguas residuales |
| 1980 | Fukami, Mizumoto y Tanaka | Inferencia condicional borrosa |
| 1983 | Hirota & Pedrycz | Conjuntos difusos probabilísticos |
| 1983 | Takagi & Sugeno | Derivación de reglas de control difuso |
| 1983 | Yasunobu, Miyamoto et al. | Control difuso predictivo |
| 1984 | FIP'84 (Hawaii) | Se establece la IFSA |
| 1984 | Sugeno & Murakami | Control de parqueo de un carro |
| 1985 | Kiszka, Gupta et al. | Estabilidad de sistemas difusos |
| 1985 | Togai & Watanabe | Chip difuso |
| 1986 | Yamakawa | Hardware de un controlador difuso |
| 1987 | Boom Difuso | Productos de consumo en Japón |
| 1987 | Senday (Japón) | Operación del tren subterráneo de Senday |
| 1988 | Dubois & Prade | Razonamiento aproximado |
| 1989 | MITI (Japón) | SE inaugura LIFE |
| 1989 | IFSA (Japón) | Se crea SOFT |
| 1990 | Lee | Lógica difusa en control de sistemas |
| 1990 | Matsushita Electric Ind. | Control difuso de una lavadora |
| 1991 | Mitsubishi Electric Corp. | Video conferencia aplicando inferencia difusa |

Nota: Matsushita Electric Industrial Co. Ltda. es el nombre dado a Panasonic en Japón

La primera compañía en aplicar lógica difusa a un producto de consumo fue Matsushita Electronics Industrial en 1987 para una ducha en la cual se controlaba la temperatura del agua por medio de un controlador difuso. Otro hecho importante para el control difuso fue el desarrollo del chip analógico para hacer inferencias difusas desarrollado por Yamakawa.

Veamos a continuación algunos desarrollos realizados en diferentes partes del mundo. Destacamos los hechos por OMRON que es una de las empresas que han desarrollado y aplicado la tecnología difusa en Japón. También encontramos en Japón empresas que han generado aplicaciones de lógica difusa y redes neuronales en equipos de procesamiento de imágenes y productos de consumo; estas empresas son: Canon, Hitachi, Matsushita Electric, Minolta, Mitsubishi, Ricoh, Sanyo, y Sharp. Es importante resaltar el aporte de LIFE una de las asociaciones más grandes para el desarrollo de aplicaciones difusas fundada en 1989 por la industria y el gobierno Japonés. En Estados Unidos encontramos aplicaciones como el control supervisorio difuso para un controlador PID en el control de un motor de la General Electric's, aplicaciones en robots móviles, chip VLSI de inferencia difusa, lógica difusa en sensores de procesamiento para sistemas de manufactura, en aplicaciones espaciales, y en aires acondicionados. En Europa encontramos el gran boom que se dio en Alemania.

Son importantes los aportes hechos por asociaciones internacionales como la IEEE, LIFE, IFSA, EURIT, NAFIPS, SAFLT, que a través de sus congresos han permitido la difusión de la aplicación de la teoría de conjuntos difusos y la lógica difusa a la ingeniería de control.

En la actualidad el control difuso es una de las áreas de control inteligente con más auge como se puede comprobar en la gran cantidad de publicaciones recientes de libros y artículos sobre el tema. También podemos destacar que en la actualidad a través de Internet, el adelanto tecnológico que ha revolucionado las comunicaciones, podemos

obtener cursos sobre control difuso con su respectiva certificación o material informativo que incluye software y hardware difuso de tipo comercial o educativo.

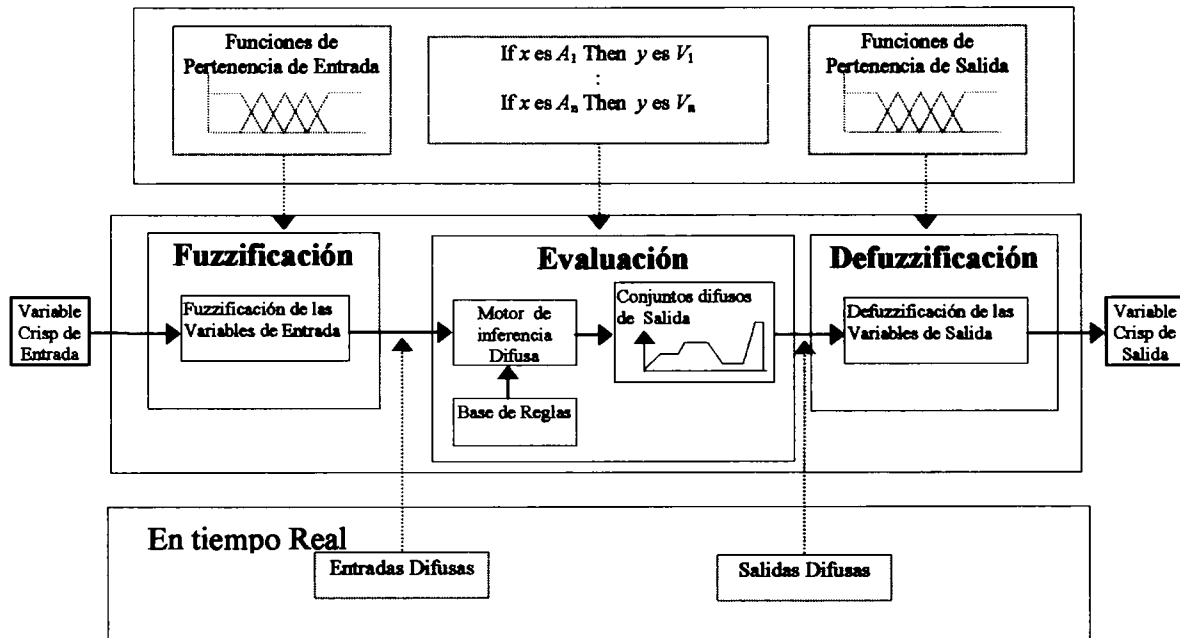


Figura 4-1 Estructura de un Controlador difuso CD

4.2. ESTRUCTURA BÁSICA DE UN CONTROLADOR DIFUSO

En las diferentes publicaciones como ya lo hemos anotado en el capítulo 3 el modelo de Mandani o sistema con fuzzificación y defuzzificación ha sido el más aceptado para la implementación de controladores difusos conocidos como CD (Fuzzy Logic Controller), esto debido a que es un modelo muy intuitivo y se presta para incorporar mejor una estrategia de control lingüística gracias a sus interfaces de fuzzificación y defuzzificación. Por este motivo es tomado como modelo básico. La figura 4-1 describe la estructura de un CD de este tipo, la figura 4-2 muestra la aplicación de un CD al control de un proceso.

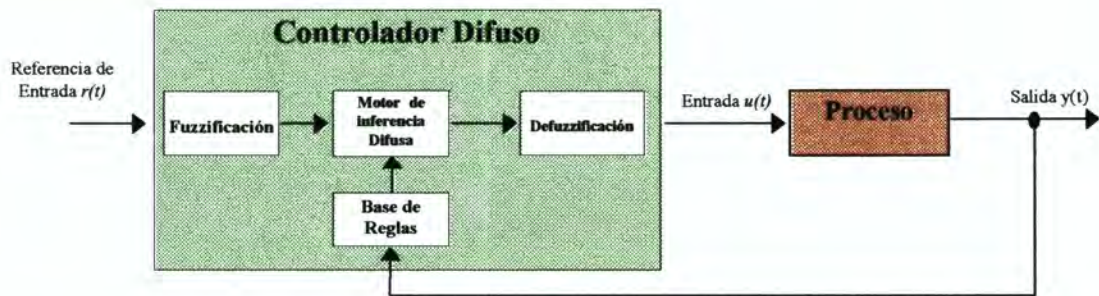


Figura 4-2 Aplicación de un Controlador difuso CD al control de un proceso

4.2.1. Interface de fuzzificación como anota Lee,¹³ la interface de fuzzificación involucra tres funciones así:

- 📖 Mide las variables de entrada
- 📖 Lleva a cabo un mapeo a escala que transfiere el rango de valores de las variables de entrada en su correspondiente universo de discurso.
- 📖 Ejecuta la función de fuzzificación que convierte datos de entrada en los apropiados valores lingüísticos los cuales pueden ser vistos como etiquetas de conjuntos difusos.

Podemos ver la interface de fuzzificación como una caja negra que recibe valores crisp (valores reales) propios de las variables de estado del proceso controlado y que a través de ella son transformados de escala a un universo de discurso, la salida de la caja negra será un conjunto difuso con su respectivo grado de pertenencia al mismo, el cual es compatible con

¹³ LEE, Op. Cit., p.

los antecedentes de las reglas. En algunas aplicaciones de CD se utiliza una normalización previa de la fuzzificación. En el manual del Fuzzy Logic Toolbox de MatLab encontramos una definición que nos puede dar una referencia para entender este concepto: *"la fuzzificación realmente no es más que una tabla de búsqueda o la evaluación de una función"*, y esto tal vez porque el resultado de la fuzzificación de un valor crisp (número real) es un conjunto difuso y su grado de pertenencia $\mu(x)$ a éste. Veamos un ejemplo:

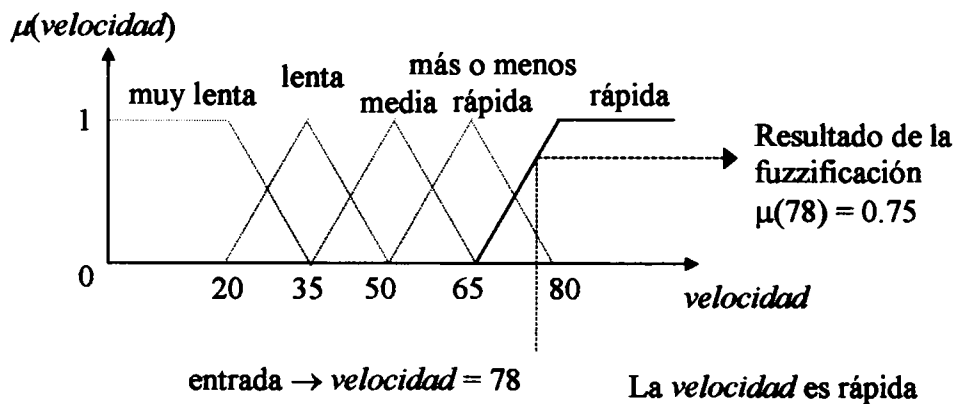


Figura 4-3 Proceso de fuzzificación para la variable *velocidad*

Del ejemplo de la figura 4-3 podemos decir que la variable de estado *velocidad* es *rápida* en un grado de 0.75, debemos tener en cuenta que cada entrada es fuzzificada sobre todas las funciones de pertenencia requeridas por las reglas. De tal manera que si para el mismo ejemplo en una regla de la base de reglas se tuviera en el antecedente la etiqueta *lenta* la fuzzificación daría cero para la pertenencia a este conjunto en el estado actual de la variable *velocidad*.

4.2.2. Base de conocimiento comprende un conocimiento del dominio de la aplicación y el servicio del objetivo de control. La cual involucra una "base de datos" y una "base de reglas de control lingüísticas (difusa)."

- ☐ La base de datos proporciona necesariamente definiciones, que son usadas para definir reglas de control lingüísticas y manipulación de datos difusos en un CD,
- ☐ la base de reglas caracteriza el objetivo de control y la política de control del dominio del experto por medio de un conjunto de reglas de control lingüísticas.

Lo que se quiere representar con la base de conocimiento es la información proveniente de la heurística del experto necesaria para el buen funcionamiento de las interfaces de fuzzificación y defuzzificación, y la base de reglas, lo que incluye los conjuntos difusos de entrada y de salida, básicamente en la definición de sus funciones de pertenencia con el fin de que puedan representar los valores lingüístico de las variables de estado del proceso, los dominios físicos y los universos de discurso. En el caso de usar normalizaciones los parámetros de éstas, al igual que si se han de discretizar los dominios continuos de las variables de estado.

Con la base de reglas se representa de manera estructurada la estrategia de control lingüística del operador experto, como un conjunto de reglas **If-Then**

If (estado del proceso) Then (salida de control)

como lo vimos en el capítulo anterior la parte **If** representa el antecedente de la regla, y es una descripción difusa del estado del proceso en términos de una combinación lógica de proposiciones difusas. La parte **Then** es llamada consecuente y que al igual que el antecedente es una combinación lógica de proposiciones difusas que dan como resultado los valores lingüísticos que contribuirán a la acción final de control.

Es importante al hablar de la base de reglas en un CD definir la “**Completes**” como la propiedad que tiene un algoritmo de control para que siempre sea capaz de inferir una apropiada acción de control para todos los estados del proceso.

Veamos un ejemplo de una base de reglas que tiene como variables de entrada el error y la derivada del error. Por simplicidad mostramos sólo diez reglas, pero en realidad la base está compuesta por cuarenta y nueve reglas:

1. If (error is NL) and (de is NL) then (out is NX) (1)
2. If (error is NL) and (de is NM) then (out is NX) (1)
3. If (error is NL) and (de is NS) then (out is NX) (1)
4. If (error is NL) and (de is ZE) then (out is NX) (1)
5. If (error is NL) and (de is PS) then (out is NX) (1)
6. If (error is NL) and (de is PM) then (out is NX) (1)
7. If (error is NL) and (de is PL) then (out is NX) (1)
8. If (error is NM) and (de is NL) then (out is NX) (1)
9. If (error is NM) and (de is NM) then (out is NX) (1)
10. If (error is NM) and (de is NS) then (out is NL) (1)

Dado que la anterior estructura es algo difícil de manejar para sistemas con muchos conjuntos difusos (particiones difusas) es común y útil utilizar un arreglo matricial como se muestra a continuación.

Tabla No 4-2 Arreglo FAM

| des | NL | NM | NS | ZE | PS | PM | PL |
|-----|----|----|----|----|----|----|----|
| NL | NX | NX | NX | NX | NS | ZE | PX |
| NM | NX | NX | NL | NL | NS | PS | PX |
| NS | NX | NL | NL | NM | PS | PS | PX |
| ZE | NX | NL | NL | ZE | PL | PL | PX |
| PS | NX | NS | NS | PM | PL | PL | PX |
| PM | NX | NS | PS | PL | PL | PX | PX |
| PL | NX | ZE | PS | PX | PX | PX | PX |

A este formato se le conoce como FAM por sus siglas en ingles (matriz difusa de memoria asociativa). En este arreglo la primera columna y la primera fila representan todos los conjuntos difusos de la dos variables de entrada (error y derivada del error), las demás casillas representa el valor que toma la variable de salida de acuerdo a la combinación de las filas y columnas respectivas. Es de notarse que en el arreglo FAM se encuentran todas las cuarenta y nueve reglas del controlador.

4.2.3. El motor de inferencia el motor de inferencia es llamado por Lee en su artículo "*Fuzzy Systems: Logic in Controller*" el núcleo del CD, ya que tiene la capacidad de simular los mecanismos de decisión humana basado en conceptos difusos y de inferir acciones de control difuso empleando implicaciones difusas y las reglas de inferencia de la lógica difusa.

Como ya se había mencionado en el capítulo 3 el motor de inferencia usa principios de lógica difusa para combinar las reglas difusas **if-then** en la base de reglas en un mapeo de los conjuntos difusos de entrada a conjuntos difusos de las salidas.

El motor de inferencia o sistema de activación de reglas, puede ser de dos tipos:

Inferencia basada en una composición: en este caso, las relaciones difusas representando el sentido de cada regla individual se agregan en una relación difusa que describe el sentido del conjunto completo de reglas. Entonces la inferencia o activación con esta relación difusa se realiza mediante la operación de composición entre la entrada crisp fuzzificada y la relación difusa que da el significado del conjunto completo de reglas. Como resultado de la composición se obtiene el conjunto difuso que describe el valor difuso de la salida de control global.

Inferencia basada en cada regla individual: en este caso, se activa primero cada regla aislada. Esta activación puede describirse fácilmente:

- ❑ Calculando el grado acoplamiento entre la entrada crisp y los conjuntos difusos que describen el significado del antecedente de la regla.
- ❑ Truncando el conjunto difuso que describe el significado del consecuente por el grado al que el antecedente de la regla se ha acoplado con la entrada crisp. Finalmente, los valores truncados de la salida de cada regla se agregan, obteniendo la salida global.

Usualmente se prefiere el segundo tipo de inferencia porque es muy eficiente computacionalmente, y ahorra mucha memoria.

4.2.4 Interfaces de defuzzificación debido a que los sistemas de control requieren valores numéricos para poder modificar la dinámica de los sistemas, se requiere entonces que la información lingüística contenida en los conjuntos difusos de salida sea convertida en valores numéricos que puedan ser interpretados por los actuadores (niveles de voltajes, presión o corriente, etc.) este proceso lo realiza la interface de defuzzificación.

Los más importantes procedimientos de defuzzificación son los cinco siguientes definidos por Verdegay¹⁴: Centro de gravedad, centro de las sumas, centro de las mayores áreas, primer máximo, y elección por la altura.

Supongamos un conjunto de M reglas

$$\text{If } x_1 \text{ es } F_1^{(l)} \text{ and ... and } x_n \text{ es } F_n^{(l)} \text{ Then } u \text{ es } U^{(l)}, l = 1, \dots, M \quad (4.1)$$

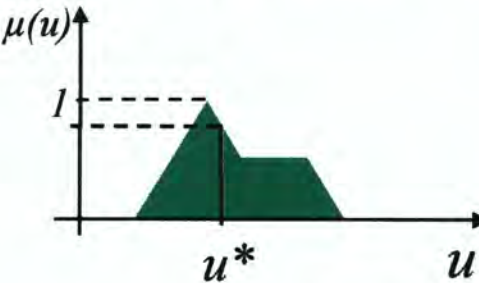
El resultado de activar, estas reglas mediante valores numéricos $\underline{x} = (x_1, \dots, x_n)^T \in U$ serán conjuntos difuso truncados notados por $\mu_U^{(1)}, \dots, \mu_U^{(M)}$. Finalmente, el control global μ_U se obtiene por unión de esos conjuntos difusos truncados, y a partir de él es de donde hay que extraer un único valor (según el procedimiento de defuzzificación que se elija).

Nótese que la altura de $\mu_U^{(k)}$ es igual al grado de acoplamiento del antecedente de la k -ésima regla, y se notara por f_k . El pico (la moda) de $\mu_U^{(k)}$ es igual a la moda de su versión

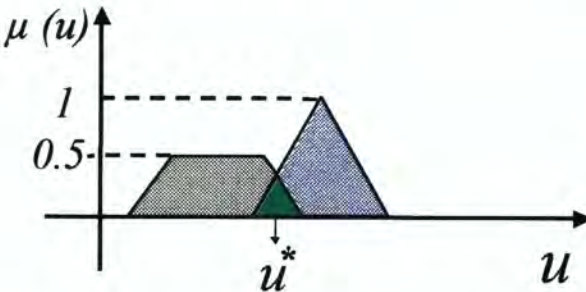
¹⁴ VERDEGAY, José L. "Introducción al Control Difuso" En Memorias Primer Congreso de la ACA, Noviembre de 1994

no truncada (si la función de pertenencia es triangular, el pico corresponde al valor del soporte que tenía grado de pertenencia 1).

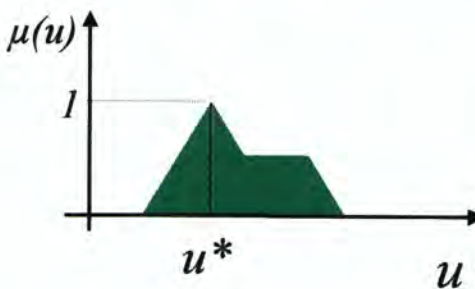
Método del centro de gravedad (centroide o centro de área): en el caso discreto,

$$u^* = \frac{\sum_{i=1}^l u_i \cdot \mu_U(u_i)}{\sum_{i=1}^l \mu_U(u_i)} = \frac{\sum_{i=1}^l u_i \cdot \text{Max}_k \mu_{CLU^{(k)}}(u_i)}{\sum_{i=1}^l \text{Max}_k \mu_{CLU^{(k)}}(u_i)}$$


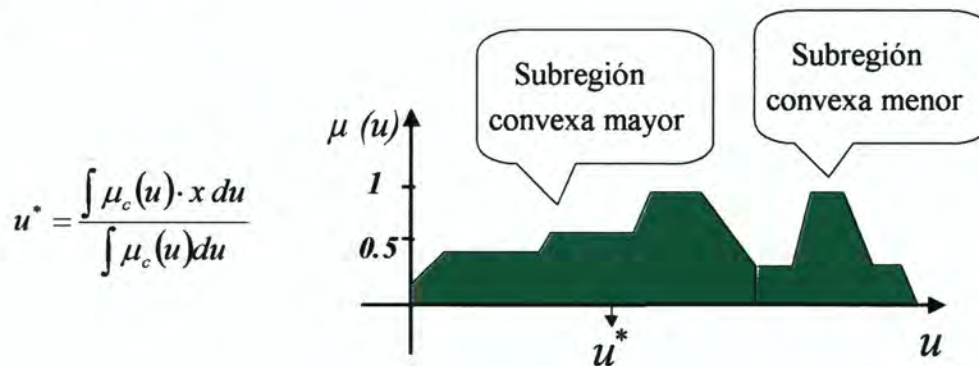
Método del centro de las sumas: es un método similar al centro de gravedad pero más rápido,

$$u^* = \frac{\sum_{i=1}^l u_i \cdot \sum_{k=1}^n \mu_{CLU^{(k)}}(u_i)}{\sum_{i=1}^l \sum_{k=1}^n \mu_{CLU^{(k)}}(u_i)}$$


Defuzzificación por método de la altura (máxima pertenencia o método del mayor): si $c^{(k)}$ nota el valor pico de μ_{LU} , y f_k es la altura de μ_{CLU} , entonces

$$u^* = \frac{\sum_{k=1}^m c^{(k)} \cdot f_k}{\sum_{k=1}^n f_k}$$


Método del centro del área mayor : si el conjunto difuso de salida tiene al menos dos subregiones convexas, entonces u^* se calcula usando el método del centroide de la subregión de área mayor

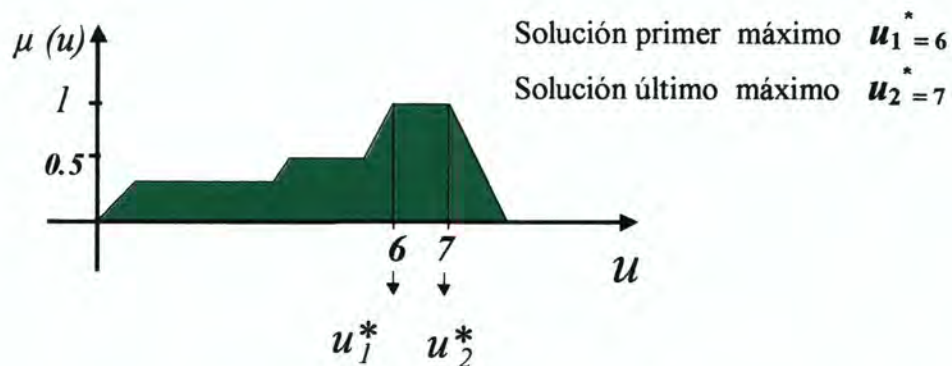


Defuzzificación por el primer máximo:

$$u^* = \text{Inf} \{u / \mu_U(u) = \text{hgt}(U)\}$$

También se puede calcular el **último de los máximos** como

$$u^* = \text{Sup} \{u / \mu_U(u) = \text{hgt}(U)\}$$



Aplicaciones: hacemos un resumen de algunas aplicaciones típicas en control que se suman a las ya mencionadas en el capítulo 3 (tabla 3-1); estas son aires acondicionados, hornos microondas, operación automática de trenes, reactor nuclear, dispositivos de memoria difusa y las listadas en la tabla 4-3.

Tabla 4-3 Aplicaciones de Control Difuso

| Aplicaciones | Descripción |
|----------------------|---|
| Cámaras | De entre diversos puntos de foco se escoge a través de reglas if-then el foco óptimo |
| Video Cámaras | En la estabilización de imágenes |
| Lavadoras | Mediante sensores se determina que tan sucia está el agua y como se va limpiando para determinar el ciclo de lavado apropiado |
| Fotocopiadoras | Control electrográfico y calidad de impresión |
| Automóviles | En el sistema de transmisión |
| Grúas Automáticas | En el cargue y descargue automático de containers en buques |
| Control de trenes | CSC Controlador de velocidad constante y TASC Control automático de parada para el tren |
| Televisores | Control de la calidad de la imagen |
| Aires Acondicionados | Se evitan fluctuaciones bruscas en la temperatura |

Es claro que la lógica difusa constituye un parte principal en el modelamiento de una adecuada estrategia de control lingüística o basada en reglas, independiente del alcance y/o implementación, la cual sirve de puente entre el procesamiento simbólico (propio de la heurística) y la computación numérica.

4.3. CONTROL DIFUSO DIRECTO (CDD)

El CDD es el más popular, siendo usado por muchos investigadores a través del mundo y es ampliamente conocido. El objetivo fundamental de esta estructura es el de generar una acción apropiada de control basado en la señal de error luego de la realimentación de la salida del proceso. La arquitectura del CDD es mostrada en la Figura 4.4. La entrada al sistema es la señal de referencia R . En la literatura se encuentran infinidad de ejemplos utilizando esta arquitectura.

El controlador difuso actúa directamente sobre la planta o el proceso a controlar.

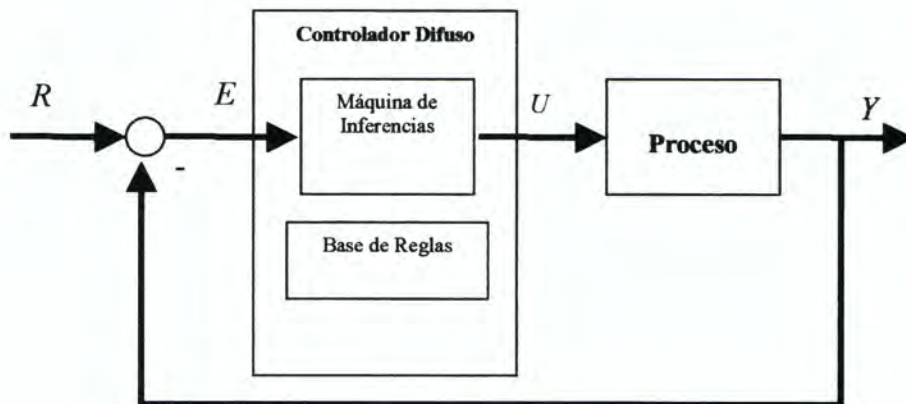


Figura 4-4 Controlador difuso directo (CDD)

4.4. CONTROL PID DIFUSO

El controlador PID difuso permite el control de sistemas no lineales que con el PID convencional serían imposibles de controlar.

Las reglas lingüísticas para un controlador PID difuso son de la forma:

If e es (....) And Δe es (....) And δe es (....) Then Δu es (....)

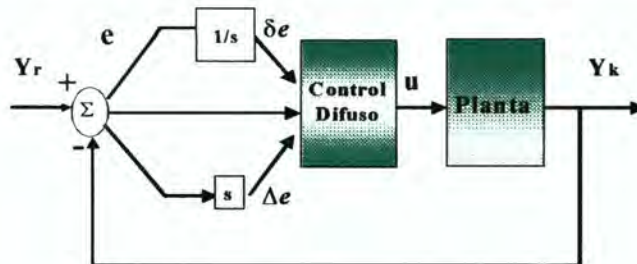


Figura 4-5 Controlador PID difuso

Las variables de estado del proceso, las cuales representan el contenido de los antecedentes de las reglas difusas son:

- ❏ *Error (e)*
- ❏ *Incremento en el error (Δe)*
- ❏ *Suma de los errores (δe)*

Las variables de control, sustentan los consecuentes de la base de reglas difusas, ellas pueden ser:

- ❏ *Señal de control (u)*
- ❏ *Incremento en la señal de control (Δu)*

La arquitectura básica de un controlador PID difuso se muestra en la figura 4-5.

Elaboración de las reglas de control: un controlador PID difuso con siete conjuntos difusos por variable, requiere de 343 reglas de control.

En la práctica es más fácil encontrar controladores difusos PI y PD ya que con igual cantidad de conjuntos difusos sólo requieren de 49 reglas de control.

Por simplicidad examinaremos las reglas para un controlador difuso PI, cuyas entradas son el error y el cambio en el error (e y Δe) Su salida será el incremento en la señal de control (Δu).

Análogamente que en un controlador convencional discreto se tiene que:

Se define al error como $e(k) = Yr(k) - Yk(k)$

Entonces $\Delta e(k) = e(k) - e(k-1)$, de igual forma $\Delta u = u(k) - u(k-1)$, y

$$\delta e(k) = e(k) + e(k-1)$$

Las reglas de control serán de la forma:

If e es (....) and Δe es (....) then Δu es (....)

En la figura 6-10 se muestra la estructura del sistema de lógica difusa. Se observan dos interfaces de entrada que fuzzifican las dos variables de entrada "aceleración" y "speed_error".

El bloque de reglas en la unidad, contiene todas las reglas difusas que representan la estrategia de control y al lado derecho está la variable de salida "PMV_Set_Value", la cual está defuzzificada en una interface de salida.

En la figura 6-11 se muestran las variables lingüísticas y sus funciones de pertenencia. Ellas están conectadas por un total de 12 reglas difusas.

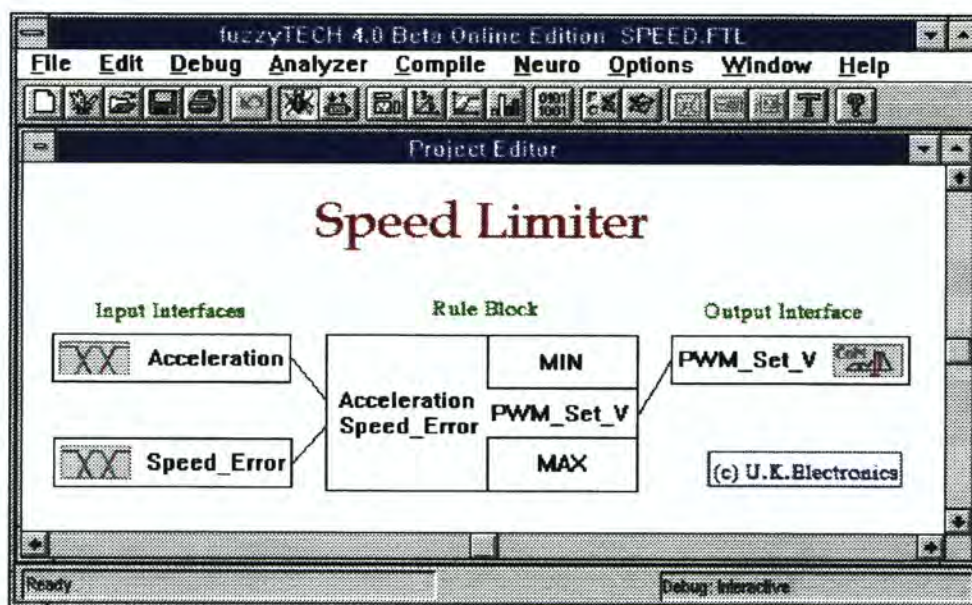


Figura 6-10 Estructura del sistema de lógica difusa

Todas las funciones de pertenencia son de tipo estándar (triangulares y trapezoidales).

Como método de defuzzificación se usa el del centro del máximo. Luego de la

optimización de las estrategias de las reglas difusas sobre los diferentes camiones y varias condiciones de carga y terreno, el limitador de velocidad mostró una curva de respuesta mucho más suave y proporcionando una gran precisión para mantener la velocidad límite comparado con los controladores convencionales. (ver figura 6-12)

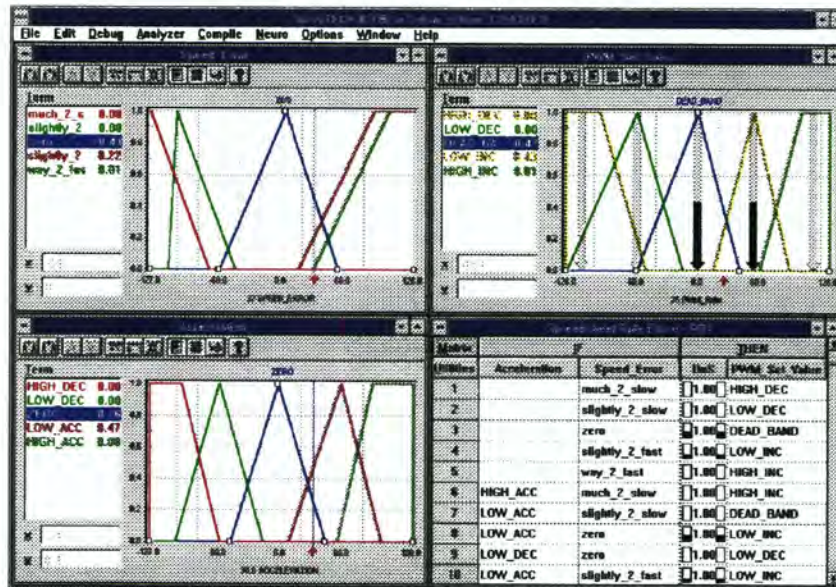


Figura 6-11 Variables lingüísticas y funciones de pertenencia.

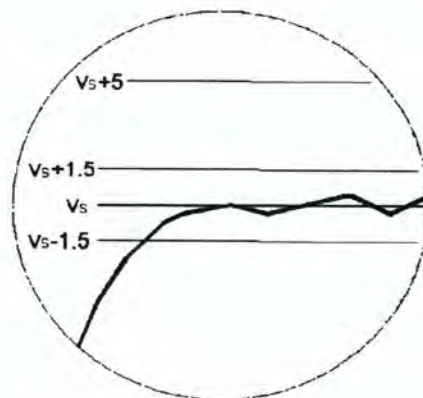


Figura 6-12 Curva de respuesta

7. HERRAMIENTAS DE SOFTWARE PARA DISEÑO, SIMULACIÓN E IMPLEMENTACION DE CONTROLADORES DIFUSOS

Debido al desarrollo que ha tenido en los últimos años las aplicaciones de lógica difusa al control y la necesidad de herramientas que permitan implementar dichas aplicaciones han surgido diferentes herramientas de software que ayudan a diseñar, simular e implementar controladores difusos. Estas herramientas las podemos resumir en la siguiente tabla

Tabla 7-1 Herramientas de Software

| Software | Diseño |
|---|---------------------------|
| CubiCalc | HyperLogic Corp. |
| TilShell | Togai Infalogic |
| <i>fuzzyTech</i> | INFORM |
| FIDE | Aptronix |
| RT/Fuzzy | Integrated Systems Inc. |
| Fuzzy Knowledge Bulder o Manifold Graphics Editor | Fuzzy Systems Engineering |
| Fuzz-C | Byte Craft Ltd. |
| <i>Fuzzy Logic</i> ToolBox para MatLab | Math Works Inc. |
| FS-10AT | Omron |
| FuzzySoft | GTS Trautzi GMBH |
| LFLC Plus 1.2 | Inst. de Geonics |
| ES/Kernel | Hitachi Corp. |
| VisSim ver. 1.2 | Visual Solutions Inc. |
| FuziCalc ver 1.0 | FuziWare, Inc. |
| KBG para MC68HC11 y MCH68HC05 | Motorola Inc. |
| Fuzzy Logic Development Kit (FULDEK) | TSI Enterprises Inc. |
| Fuzzy Logic Designer ver. 1.0 | Byte Dynamics Inc. |

Todas estas herramientas de software permiten al usuario fácilmente construir una base de reglas, definir funciones de pertenencia, analizar gráficamente la base de reglas y las salidas del sistema a implementar. Algunas de estas herramientas permiten generar código fuente o código ensamblador que implementa la base de reglas, y que puede ser llevado a un microcontrolador o un PLC para definir una estrategia de control difuso.

El siguiente esquema permite ver donde interviene el software de diseño difuso dentro de la implementación de un controlador difuso.

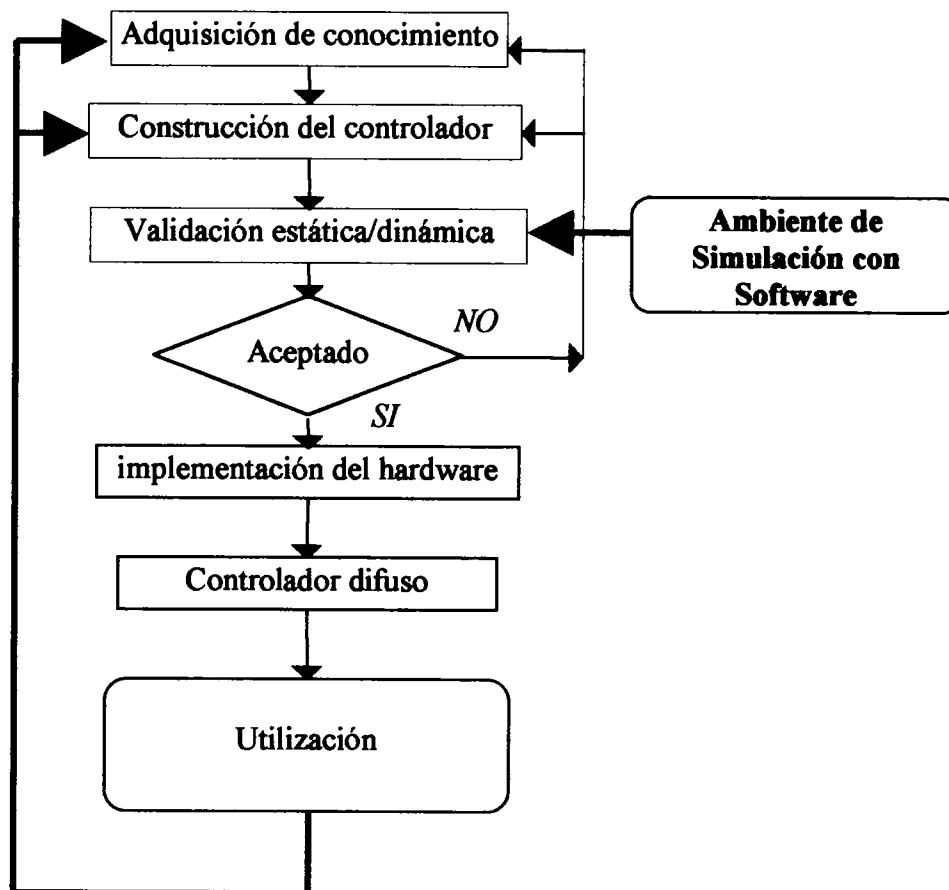


Figura No 7-1 Desarrollo de un controlador difuso (FLC)

La parte del diseño de un FLC que define la aprobación de sus propiedades dinámicas y estáticas es muy exigente porque requiere experimentos con el controlador trabajando bajo condiciones regulares de funcionamiento o en un ambiente de simulación apropiado. Esto es principalmente para analizar aspectos de la coherencia lógica de la estructura del controlador, exactitud de las propiedades de mapeo estático, consistencia de las reglas y el análisis del desempeño del controlador en una estructura de lazo cerrado. Todo esto debido a que partes del conocimiento del experto puede estar incompleto o estar en conflicto. Es allí donde las herramientas de software ayudan a implementar dicha estrategia de control mediante la depuración y validación del modelo.

Estas herramientas difieren unas de otras en el tipo de código que pueden generar y varían en la forma de editar la base de reglas, funciones de pertenencia y su capacidad de simulación y depuración.

A continuación haremos referencia de estas herramientas de software que provienen de la bibliografía en algunos casos y en otros del análisis de demos conseguidos a través de Internet ya que la mayoría de estos programas son de uso comercial e involucran costos que no están al alcance de este trabajo pero sí dejamos referencia de la información para quien esté interesado en profundizar sobre ellas.

Se aclara que el análisis hecho de las características de CubiCalc 2.0, TILShell 3.0, y FIDE 2.0 provienen de versiones beta que pueden cambiar en la versión comercial final de estos programas.

CUBICALC (V. 2.0)

Este es un programa desarrollado por HyperLogic Corp. el cual trabaja bajo el entorno Microsoft Windows. CUBICALC está implementado alrededor de un sistema experto que tiene una interface intuitiva que hace que el programa sea fácilmente comprensible para cualquier usuario, sin necesidad de tener un conocimiento muy técnico. Es fácil de usar y posee características convenientes y un bajo costo por lo que es quizás el programa de lógica difusa más usado en los Estado Unidos.

CUBICALC posee un editor de texto para ingresar las reglas como frases if-then y un editor gráfico para definir las funciones de pertenencia como una unión de líneas segmentadas. Se pueden usar pesos para las reglas, además soporta modificadores lingüísticos (hedges) en las reglas. Vemos un ejemplo:

```
(HiRuleWeight) If (Temperatura es algo caliente o la Humedad es alta) and
Temperatura_cambio is incremetado
    then Potencia debería ser Alta
    @HiRuleActv
```

El peso de la regla puede ser especificado como una variable y ser usado iterativamente por el usuario o por otro módulo de computación. La frase opcional " @HiRuleActv" asigna el nivel de activación para una variable llamada HiRuleActv; esta variable puede entonces ser examinada para propósitos de depuración o por otro módulo de cálculo.

Cuando se está definiendo una nueva función de pertenencia, el editor gráfico presentará primero por defecto una simple figura formada por unos vértices conectados. El usuario puede modificar la forma de las funciones de pertenencia moviendo los vértices con el mouse o adicionando nuevos vértices para crear formas más complejas. Esta manera de editar las funciones de pertenencia es aplicada en la mayoría de programas para control difuso como en el ToolBox de Matlab. Una vez el término difuso usado en el texto de la regla es definido, las reglas pueden ser compiladas y ejecutadas en el entorno del CubiCalc. Para probar las reglas, el usuario puede iterativamente ingresar valores de entrada o seleccionar al CubiCalc para que itere sobre un rango de valores de entrada, o sobre datos de un archivo. Los resultados pueden ser mostrados como un texto o un ploteo. Alternativamente, CubiCalc puede evaluar expresiones numéricas definidas por el usuario para calcular los valores de entrada o la base de reglas. Esta capacidad permite al usuario simular la respuesta del sistema a la decisión difusa y probar la base de reglas dinámicamente. Por ejemplo, supongamos que la base de reglas contiene la temperatura como una variable de entrada. Entonces el usuario puede crear un bloque de expresiones que pueden contener la frase

$$\text{Temperatura} = \text{Temperatura} + (k * \text{Potencia}) - \text{perdidas};$$

Así, para cada iteración de prueba el valor de entrada de la temperatura es actualizado basado en una temperatura previa y los valores de la potencia, efectivamente simulando la respuesta de una planta dinámica. Además para definir un bloque de expresiones como el modelo de simulación, el usuario puede también especificar un bloque de expresiones para ser evaluadas inmediatamente antes de la inferencia y un bloque de expresiones para ser

evaluadas inmediatamente después de la inferencia, para un procesamiento anterior y posterior de las variables. Estas expresiones numéricas son interpretadas durante la simulación y debe conformar una sintaxis como C o Pascal.

CubiCalc soporta construcciones condicionales e iterativas, tan satisfactorias como una función matemática que igual incluye aritmética de matrices.

Cuando probamos la base de reglas, cualquier variable habilitada para CubiCalc (entradas, salidas, variables de activación de reglas o variables usadas en una expresión numérica) puede ser ploteada en un plano de dos dimensiones, o en un ploteo polar. La composición de las funciones de pertenencia (como la forma de onda a ser defuzzificada) también se puede mostrar.

Para ayudar a entender la función implementada por la base de reglas, CubiCalc puede mostrar un ploteo tridimensional de la superficie de decisión. Este ploteo permite al usuario ver las variables de salida como una función de cualquiera de dos variables de entrada. Secciones a través de la superficie de decisión, muestran las variables de salida como una función de solamente una variable de entrada, puede también ser mostrado por la selección de un punto en la superficie de decisión.

Muchos programas difusos pueden mostrar un ploteo de la superficie de decisión comprendiendo las variables de entrada y salida de las reglas. Sin embargo, CubiCalc puede mostrar una superficie de ploteo incluyendo cualquier variable habilitada para esto; así las variables de entrada para la superficie de decisión pueden ser datos de entrada antes

de la fase del procesamiento numérico y las variables de salida pueden representar datos de salida después de la fase de postprocesamiento numérico.

Cuando el usuario está satisfecho con las pruebas de la base de reglas, el generador de código de CubiCalc puede trasladar una base de reglas en un archivo para ser usado con una función de inferencia de una librería en C. El código fuente para la librería es habilitado como un producto separado para diferentes plataformas.

CubiCalc también soporta el protocolo Windows Dynamic Data Exchange(DDE), el cual permite el intercambio entre aplicaciones Windows. Una completa simulación puede ser desarrollada como una aplicación separada (programa) que use DDE para llamar CubiCalc para la ejecución de reglas. Otras aplicaciones, tal como hojas electrónicas y programas de bases de datos, también puede usarse las características DDE para acceder el motor de computación difusa CubiCalc.

CubiCalc tiene un entorno versátil que provee e integra bien la lógica difusa, el procesamiento numérico, y capacidades gráficas para explorar diversas aplicaciones. La habilidad para examinar cualquier variable en el sistema y para incluir variables en una computación o ploteo provee gran flexibilidad. Sin embargo, la capacidad de depuramiento de la base de reglas en CubiCalc es un poco inconveniente, por que los niveles de activación de reglas son explícitamente asignados a variables y rastreados explícitamente.

Existen otras herramientas de software que proveen métodos más automáticos para guardar rastreos de la actividad de la base de reglas. A pesar de ello, CubiCalc ofrece un entorno exploratorio extremadamente rico que es simple de usar a un costo notablemente bajo.

TILSHELL (V. 3.0)

Este producto es de Togai InfraLogic's, trabaja bajo ambiente Windows. TilShell es una herramienta para ingeniería de control difuso con un entorno CASE (Computer Aided Software Engineering). El principal objetivo del TilShell es proveer un entorno gráfico para la creación de un archivo de lenguaje de programación difusa (FPL); FPL es un lenguaje para la descripción de una base de reglas que sirve de interface común entre los productos de Togai's. Se puede aplicar un rango de generadores de código a un archivo FPL para producir en línea un completo código fuente C, código ensamblador para Togai's FC110 (procesador difuso), o código ensamblador para varios microcontroladores como Hitachi, Mitsubishi, Oki, Intel, y Motorola.

En el entorno TilShell, cada variable de entrada y salida es representada como un objeto gráfico; el bloque de reglas relaciona las entradas con las salidas y es también representado como un objeto gráfico. El usuario empieza por crear estos objetos y conectar estos por arcos para especificar sus dependencias.

Diferentes métodos de defuzzificación pueden ser fijados para cada objeto de salida.

Dando doble click en una objeto de entrada o salida podemos abrir un editor para definir las funciones de pertenencia asociadas con la variable de entrada o salida. Dentro del editor, una función de pertenencia puede ser definida gráficamente por puntos hechos con click del mouse o textualmente por ecuaciones matemáticas; allí se encuentra también un comando rápido para creación de un conjunto completo de funciones de pertenencia uniformemente espaciadas. Una ventana de edición gráfica muestra todas las funciones de pertenencia correctamente definidas para la variable de entrada o salida asociada; durante la prueba de la base de reglas, la ventana de edición sirve como una ventana de depuración por pantallas adicionales de los valores de cada función de pertenencia.

Dando doble click en un bloque-objeto base de reglas se abrirá el editor de reglas. Existen dos tipos de editores de reglas: un editor matriz y un editor de hoja electrónica; el usuario puede seleccionar uno u otro de acuerdo a su preferencia. El editor de matriz muestra una grilla de celdas que representan cada una de las posible combinación de entrada.

El peso de una regla puede también ser asociado con cada celda. Durante la prueba de la base de reglas, el editor de matriz puede servir como una ventana de depuración mostrando en cada celda el nivel de activación de la correspondiente regla. El editor de matriz puede crear reglas que contienen solamente dos variables. Este no puede ser usado para editar una división de una matriz de reglas de una alta dimensión.

Cuando una base de reglas tiene más de dos variables de entrada, el usuario puede especificar cual de las dos combinaciones de las variables de entrada va a ser editada.

Para crear reglas que contienen más de dos variables de entradas, debería usarse el editor de hoja electrónica por su fácil manejo.

El editor de hoja electrónica puede ser visto como una típica hoja electrónica; cada fila en la hoja electrónica representa una regla, y cada columna corresponde a una variable de entrada o de salida, un operador (conectivo de la premisa), o a un peso de regla. Las reglas son creadas llenando las celdas de la hoja electrónica con el nombre apropiado de la función de pertenencia, el conectivo operador("and" para el mínimo, "or" para el máximo, "ap" para el producto aritmético, etc.) y los valores de los pesos de las reglas. Para crear una regla que no use todas las variables de entradas o salidas, las celdas correspondientes a la variable se dejan en blanco. Esta flexibilidad no está habilitada para el editor de matriz, donde toda regla representada por una matriz celda debe intrínsecamente usar todas las variables de entrada. El editor de matriz y la hoja electrónica están limitados en la expresiones de las reglas que pueden ser creadas; por ejemplo, una regla creada usando el editor de hojas electrónicas no puede mezclar operadores "and" y "or" en su premisas.

Para crear expresiones de reglas más complejas, puede ser usada una región de edición de texto en el editor de hoja electrónica para entrar reglas directamente en el formato FPL, como podemos ver a continuación:

```

RULE Rule1
  OPTIONS
    WEIGHT = 100
  END
  IF (Error IS Z) AND (dError IS Z) THEN Current = Z
END

```


Como se puede ver, el FPL es un lenguaje fácil; ya que es simple introducir reglas directamente en este formato.

El FPL soporta expresiones complejas de reglas que pueden contener más de un tipo de conectivos operadores y más de una función de pertenencia asignada a una variable. Una regla compleja puede ser mostrada en la hoja electrónica como una fila no editable.

Escribir directamente reglas en el FPL provee flexibilidades adicionales: la conclusión de una regla puede ser una expresión matemática que es evaluada en tiempo real. Por ejemplo:

```
RULE Rule1
    IF (A IS low) AND (B IS high) THEN C = 3*A + B
END
```

Este tipo de regla corresponde al propuesto por Takagi y Sugeno en 1985 como vimos en el capítulo 3; el cual permite la representación de una compleja superficie decisión con sólo un pequeño número de reglas.

Igual que la capacidad de procesamiento numérico de CubiCalc, TilShell soporta objetos de procesamiento numérico que pueden ser conectados a objetos de las salidas o entradas. Dando doble click en un objeto de procesamiento numérico se abrirá un editor de texto para especificar el cálculo numérico a ser representado.

Expresiones matemáticas y condicionales así como lazos de iteración pueden ser ejecutados por un objeto de procesamiento numérico. El lenguaje usado para el procesamiento numérico es un lenguaje como el C llamado FPL Script lenguaje. La capacidad de construir simulaciones es también posible por el FPL Script lenguaje, el cual permite al usuario definir un objeto modelo de simulación con TilShell.

Además permite graficar variables en dos dimensiones durante la simulación, Tilshell tiene amplias características para la depuración de la base de reglas y sintonización.

La prueba de la base de reglas puede hacerse a través de la entrada de un archivo del usuario o por simulación.

El usuario mediante una ventana puede hacer la sintonía de los pesos de las reglas durante una simulación y observar los efectos de estos cambios en la simulación. Al igual que también se pueden modificar las funciones de pertenencia durante la simulación. En este proceso también puede ser consultado el valor de cualquier variable. Existe también la posibilidad de hacer ploteos en tres dimensiones de la superficie de decisión. Las reglas pueden ser adicionadas o borradas; sus pesos, la asignación de funciones de pertenencia de entrada y salida, los operadores (conectivos de las premisas) son todos modificables; la forma de cualquier función de pertenencia es también modificable en línea, debido a que es actualizada automáticamente.

El TILShell soporta el Windows DDE a través del FPL Script lenguaje y hace buen uso de las características de DLL (Dynamic Link Library) de Windows, a través de DLL el usuario

puede crear sus propias funciones compiladas en C para ser llamadas en TILShell como un objeto especial de procesamiento, algoritmo de defuzzificación o como una rutina gráfica de simulación.

A continuación se muestra la pantalla del demo para DOS de Togai InfraLogic

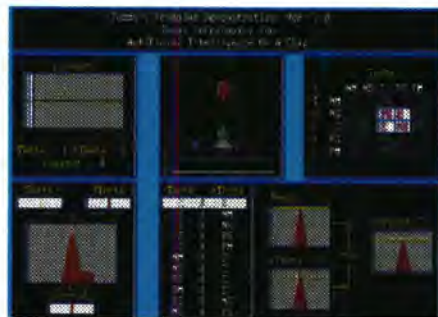


Figura 7-2 Demo del Péndulo Invertido

Togai InfraLogic también ofrece TILGen, Un producto para TILShell que automáticamente genera reglas difusas para una definición particular de un sistema difuso y un conjunto de datos de entrada y salida. TILGen usa un algoritmo basado en redes neuronales.

***fuzzyTECH* (V. 3.0 Online Edition)**

fuzzyTECH es un producto de INFORM Corp. también funciona bajo Windows y al igual que TILSHELL es una herramienta para la ingeniería de control difuso, que tiene muchas características semejantes a este. Sin embargo *fuzzyTECH* tiene un sistema diferente para el diseño de la base de reglas: en lugar de ver este como un proceso de reglas de control específico, *fuzzyTECH* ve este como un proceso de mapeos específicos en el espacio de

entrada-salida. Así las características *fuzzyTECH* involucran la construcción y sintonía de un mapeo asociativo difuso multidimensional.

El lenguaje usado por *fuzzyTECH* es llamado FTL (Fuzzy Technology Lenguaje), los archivos de este tipo permiten una interface común entre los productos de INFORM's. La generación de código en *fuzzyTECH* puede ser trasladada a un archivo FTL en código fuente en C, código ensamblador para INFORM's FUZZY-166 procesador difuso, código ensamblador para varios microcontroladores de Intel, SGS-Thomson, y Siemens, o código para PLC's producidos por Allen-Bradley, Siemens, y Klockner-Moeller (Bonn, Alemania).

En *fuzzyTECH*, un usuario empieza adicionando los nombres de las variables de entrada y salida para una lista de "variables lingüísticas". Una vez nombradas las variables, el usuario puede dar doble click en el nombre de la variable abriendo una ventana de edición gráfica por puntos para definir las funciones de pertenencias asociadas con cada variable de entrada y salida. La edición y la depuración es similar a la definición de funciones de pertenencia en TILShell pero difieren en que este no soporta la definición de ecuaciones matemáticas. También solamente tiene nueve tipos de funciones de pertenencia que pueden ser definidas para cada variable, a diferencia de otros programas que no imponen tal limitación.

Para definir un sistema difuso, el usuario debe crear objetos gráficos que representan entradas, salidas y bloque de reglas y conectar estos por arcos como en TILShell. Las entradas y salidas son objetos creados con los nombres de la lista de variables lingüísticas. Un objeto base de reglas puede ser fijado para usar diferentes operadores agregación y

composición para la inferencia, y cada objeto de salida puede ser fijado para usar diferentes métodos de defuzzificación. Las reglas en el bloque de reglas pueden ser editadas usando una hoja electrónica o un editor de matriz. En él, cada celda puede representar la combinación condición-conclusión:

If x_1 es alto y x_2 es bajo, then y es alto.

Si el peso asociado a cada celda es cero, tal regla no existe. Para una base de reglas de dos entradas y una salida con cinco funciones de pertenencia por cada variable, tendrá 125 posibles combinaciones de reglas, resultando 125 celdas organizadas en una matriz tridimensional. El usuario puede dar click en un celda de una regla y fijar su peso usando un barra desplazadora.

Existe una representación FMA (Fuzzy Associative Map) que es particularmente útil si la regla es generada automáticamente para datos de entrada salida.

fuzzyTECH puede desplegar vistas de contornos y secciones de cruce de la superficie de decisión.

La base de reglas puede ser probada por mecanismos usuales: entradas iterativas del usuario, iteración automática a través de un rango de valores de entrada, entrada de un archivo y simulación. Para una simulación debe ser creado un programa externo que se comunique con *fuzzyTECH* por medio del protocolo DDE. Durante la simulación, las variables de entradas y salidas pueden ser graficadas en función del tiempo. La depuración de toda la base de reglas y la ventana de sintonía es actualizada cada vez que se evalúa la

base de reglas. El usuario puede cambiar iterativamente los pesos de las reglas, adicionar y borrar reglas, y modificar la forma de las funciones de pertenencia durante una simulación.

fuzzyTECH también provee una facilidad en línea para depurar y sintonizar un controlador externo. Análogamente a la facilidad en línea de TILShell, este requiere una versión compilada y depurada del código de control difuso para su proceso en línea.

Un producto adicional llamado módulo NeuroFuzzy es habilitado para generar automáticamente las reglas para datos de entrada y salida. Similar al TILGen de Togai Infralogic. Este tiene tres algoritmos de aprendizaje: Backtracking Heuristics, Competitive Learning, y Variable Clustering.

fuzzyTECH soporta el concepto de operadores de parametrización lógica, donde un operador es parametrizado por un número entre 0 y 1 para proveer un comportamiento continuo. Por ejemplo, un operador con un valor de 0 puede comportarse como un operador "and"; con un valor de parámetro de 1, este puede comportarse como un operador "or"; entre 0 y 1 este tomara un comportamiento en algunas partes entre los operadores "and" y "or". Los operadores parametrizados existen a causa de que los operadores de lógica simple a menudo no reflejan cómo los humanos combinan múltiples criterios para tomar un juicio. *fuzzyTECH* ofrece tres tipos de operadores parametrizados: min-max, min-promedio y un operador de múltiple tipo llamado gamma.

Ejemplo de un bloque de reglas:

```

RULEBLOCK {
    INPUT = A , B; /* variables de entrada y salida para el bloque de reglas */
    OUTPUT = C;
    /* usa min-max operador con un valor de parámetro 0 para
       agregación la premisa valore de verdad */
    AGREGATION = (MIN_MAX, PAR(0));
    /* usa gamma operador con un valor de parámetro 0 para combinar
       niveles de activación de reglas con pesos de reglas */
    COMPOSITION = (GAMMA, PAR(0));
    /* reglas contenidas en este bloque */
    RULE {
        IF A = bajo AND B = alto THEN C = medio:
        WITH 1.0; /* esta regla tiene peso de 1.0 */
        IF A = alto AND B = alto THEN C = pequeño:
        WITH 0.8; /* esta regla tiene peso de 0.8 */
        ....
    }
}

```

Veamos a continuación las pantallas de un ejemplo del control de una grúa de container de la versión 4.0 (demo):

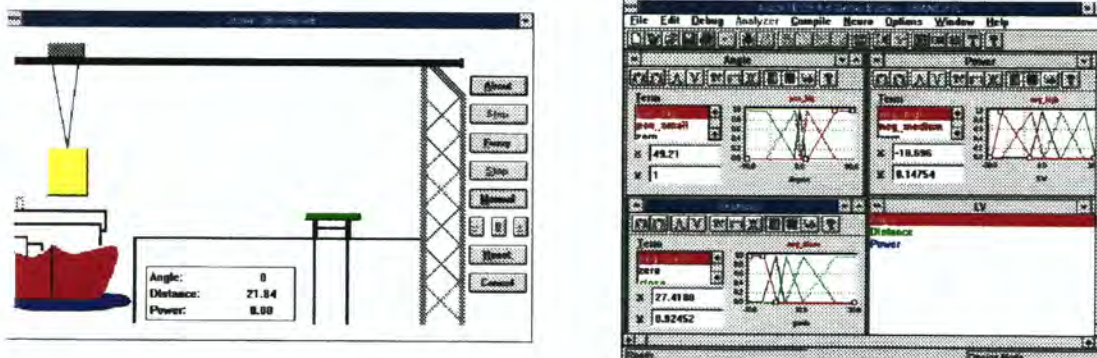


Figura 7-3 Pantallas de animación y edición *FuzzyTech*

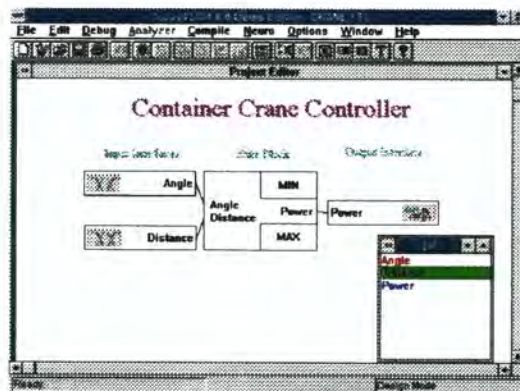


Figura 7-4 Editor de *FuzzyTech*

En las gráficas se puede apreciar el entorno gráfico para el desarrollo de sistemas difusos con *Fuzzytech*, la pantalla de animación y las funciones de pertenencia.

Este demo de la versión 4.0 tiene como ejemplos animados: el control del balanceo en un buque durante el cargue de containers, control de un disco duro, clasificación en el reciclaje de vidrio, y el control en la aplicación de cloro a una piscina pública.

FIDE (FUZZY INFERENCE DEVELOPMENT ENVIROMENT V. 2.0)

FIDE de Apronix es también una aplicación que funciona bajo Windows, y una amplia herramienta para la ingeniería de control difuso. Aunque no tiene las características compactas de TILShell o *fuzzyTech*, una versión completa de FIDE es mucho más barata.

FIDE tiene un depurador aproximadamente igual al de los dos programas antes mencionados. También puede generar código fuente en C (o con definición de archivos de datos o como un código fuente en línea), código ensamblador para un completo rango de

microcontroladores de Motorola, y código para MatLab. La posibilidad de ejecutar un módulo difuso en MatLab es particularmente conveniente para aplicaciones que involucran la integración de algoritmos matemáticos con algoritmos difusos.

FIDE proporciona un sistema de objetos similar a TILShell y fuzzyTech para representar módulos de computación. Los objetos pueden ser creados como unidades de inferencia, unidades de operaciones, y unidades de ejecución. Una unidad de inferencia representa un módulo completo de procesamiento difuso incluyendo las reglas, funciones de pertenencia, y el algoritmo de inferencia.

Una unidad de operación representa un simple módulo de computación que contiene expresiones numéricas como en C entendidas por FIDE. Una unidad de ejecución representa un módulo complejo de procesamiento que ejecuta código C del usuario; un módulo de simulación es típicamente realizado a través de una unidad de ejecución.

El proceso de desarrollo en FIDE comienza con la creación de archivos fuente individuales que definen las diferentes unidades. Un editor de texto es usado para crear los archivos fuente. Una unidad de inferencia es definida por un archivo fuente que contiene la declaración de las variables de entrada y salida, la definición de funciones de pertenencia, las frases de las reglas if-then, y las palabras claves para indicar que algoritmo de inferencia debía ser aplicado, un ejemplo del código es el siguiente:

```

fiu tvfi (Min Max) 8;
invar temperatura "gardo":30 (1) 100[
    frío (@40,0,@50,1@60,0),
    caliente(@70,0,@80,1@100,1)]
outvar fan_velocidad "rpm": 0 (5) 1000 (
    bajo = 200;

```

```

        alto = 1000);
    if temperatura is frío then fan_velocidad is baja;
    if temperatura is caliente then fan_velocidad is alta;

```

En la primera línea del código se describe que es una unidad de inferencia difusa ("fiu") (fuzzy inference unit) que emplea el algoritmo de inferencia "tvfi", usa el operador min operador para "and" y el max operador para "or", y usa una resolución de 8 bit para discretizar los valores de verdad. En la sección siguiente describe una variable llamada temperatura que acepta un rango de valores desde 30 a 100 y que puede ser discretizada en incrementos de 1; le asocia las funciones de pertenencia frío y caliente que son definidas por una lista de vértices, seguidamente define la variable "fan_velocidad" y sus funciones de pertenencia asociadas, que son singletons en este caso. Después se especifican las reglas if-then, no soporta pesos específicos para las reglas; el efecto de un peso para una regla puede ser realizado escalando la altura de la función de pertenencia de salida de la regla de acuerdo al peso deseado.

Aptronix llama a este tipo de lenguaje FIL (Fuzzy Inference Language). Un archivo FIL puede ser creado sin necesidad de que sea un editor especializado. Compilando un archivo FIL se produce una unidad de inferencia funcional que puede ser probada aisladamente. Después que un archivo FIL ha sido compilado, se puede usar un editor gráfico para modificar la forma de las funciones de pertenencia; el resultado se guarda dentro del archivo FIL.

FIDE también compila un archivo que sólo contiene la declaración de las funciones de pertenencia de las entradas y salidas. El editor de la matriz de reglas es similar al editor de

matrices de TILShell; lo cual es usado para crear reglas que contiene solamente dos variables.

FIDE permite ver un diagrama de flujo que ayuda al usuario a depurar la operación de un sistema interconectado de unidades.

RT/FUZZY (REAL-TIME FUZZY LOGIC BLOCK V. 2.4)

RT/Fuzzy arranca en estaciones de trabajo SUN, HP/Apollo y VAX, es desarrollado por Integrated Systems Inc., es mejor conocido como el fabricante de MATRIXx. Este programa tiene mucha semejanza con el programa ToolBox difuso de MatLab y Simulink, en el caso de MATRIXx este ToolBox difuso es RT/fuzzy que no es más que una herramienta para la construcción de bloques de lógica difusa. Es un módulo adicional para SystemBuild que permite la integración de elementos de procesamiento difuso en un diseño de un sistema de control general.

Un bloque difuso es creado de la misma manera que cualquier otro bloque, excepto por la forma de llenar los parámetros del bloque que es mucho más compleja.

Actualmente, una serie de formatos deben ser llenados para definir una base de reglas, con cada formato definiendo una pequeña parte de la base de reglas. El primer formato es para llenar información en general. como el nombre del bloque, número de entradas-salidas y la

metodología de inferencia deseada (conectivo, implicación, combinación de reglas y método de defuzzificación). Por ejemplo, para el campo del conectivo el usuario puede digitar "Max-Min" para especificar el operador "or" y min operador para el operador "and". Después que la información del formato es llenada, el usuario puede traer formatos adicionales para especificar cada parte de la base de reglas. Todas las variables a ser usadas en la base de reglas deben ser definidas, incluyendo entradas, salidas y variables de activación. Para cada variable de entrada o salida, el usuario debe definir un formato especificando su nombre, su tipo (crisp o difuso), su rango, y la función de pertenencia asociada con la variable. Una función de pertenencia puede ser especificada como un conjunto de punto o como una función matemática. Veamos un ejemplo del formato usado:

```
DATA TEMPERATURA;
TYPE CRISP;
RANGE [0 100];
MEMBERS FRIO [1 0.5 0 0 0];
        TIBIO EXP(-(2*X)**2);
        CALIENTE GLOBAL;
```

En este ejemplo, la función de pertenencia "frío" define números que representan valores de pertenencia para intervalos espaciados sobre el rango de las abscisas. La función de pertenencia "tibio" está definida por una ecuación. La función de pertenencia "caliente" está definida en un diccionario global. Todas la funciones de pertenencia son definidas en RT/Fuzzy sobre un rango de abscisas normalizado (para x entre -1 y +1); la curva resultante está entonces a escala de acuerdo con el rango actual definido para la variable asociada de entrada o salida.

Debido a que las variables están definidas sobre un rango normalizado y que el significado de “caliente” puede ser el mismo para diferentes variables se definen funciones de pertenencia globales como el caso de “caliente” en el código del ejemplo donde “frío” y “tibio” son declaraciones de funciones de pertenencia locales y “caliente” es obtenida por una declaración global dado que las funciones están sobre un rango normalizado. Una vez especificada cada variable, el usuario puede llamar una forma para especificar cada regla. Las reglas se digitan de la siguiente forma:

```

RULE SET_VELOCIDAD_ALTA
IF TEMPERATURA ES CALIENTE AND;
  DELTA_TEMP IS INCREMENTO;
THEN FAN_VELOCIDAD ES ALTA;
  VELOCIDAD_ALTA_ACTIV;
WEIGHT 1;

```

Debido a que cada regla es entrada como una forma separada, para cada regla debe ser dado un único nombre para que este pueda ser buscado para futuras ediciones. En la primera línea se asigna el nombre de la regla "set_velocidad_alta", seguidamente las reglas típicas if-then. Si una palabra aislada aparece en la parte de conclusión de la regla, el nivel de activación de la regla es asignado. Por eso aquí el nivel de activación de la regla es asignado a la variable "velocidad_alta_activ". Para propósitos de depuración, se debe declarar un bloque como un bloque difuso de salida. El usuario puede entrar esta información en la declaración de un formato de salida, como el siguiente:

OUTPUTS

1. FAN_VELOCIDAD
2. VELOCIDAD_ALTA_ACTIV;
3. VELOCIDAD_MED_ACTIV;
4. VELOCIDAD_BAJO_ACTIV;

Aquí el bloque difuso tendrá cuatro salidas, una para controlar el proceso y el resto para indicar el nivel de activación de tres reglas. Debido a que existen formas aisladas para cada variables de entrada, salida y activación, estas son muchas piezas para ser rastreadas. Para ayudar a rastrear la base de reglas, una ventana de Administrador de Información listará todas las variables definidas correctamente y los nombres de las reglas. Dando doble click en un nombre traerá la correspondiente forma para inspeccionar y editar.

La función de pertenencia asociada con cada variable puede también ser plotada a través del Administración de Información. La base de reglas puede ser testeada solamente por conexión de bloques difusos con otros bloques y arrancando un sistema de simulación. Comparado con otros programas RT/Fuzzy es algo complicado para construir un simple sistema difuso.

FUZZY KNOWLEDGE BUILDER (FKB V. 2.0)

Fuzzy Knowledge Builder, de Fuzzy Systems Engineering, es una herramienta basada en Windows. Este provee un entorno intuitivo para crear una base de reglas enmarcando el proceso como la creación de una superficie de decisión. Tiene amplias características para

la edición y observación de una base de reglas desde una perspectiva topológica. No es totalmente un entorno de desarrollo; es principalmente un editor de base de reglas para desarrolladores experimentados de sistemas difusos.

FKB puede trasladar una base de reglas en un archivo de alto nivel (C, Basic, y Fortran) y lenguaje ensamblador para microcontroladores (Intel, National Semiconductor, y Texas Instrument).

Es una herramienta ideal para implementar algoritmos de inferencia. Adicionalmente, la base de reglas puede ser trasladada en un archivo FPL para TILShell, un FIL para FIDE, o un archivo fuente para Fuzz-C, por lo cual puede ser usado como complemento de los anteriores programas para la edición de la base de reglas.

FKD trabaja como un programa gráfico para diseñar una superficie multidimensional; por eso fue llamado en 1994 Manifold Graphics Editor. El usuario empieza por especificar los nombres de caja de diálogos y los rangos numéricos de las variables de entrada y salida y los nombres de las funciones de pertenencia para ser usados en cada variable. Para 11 variables de entradas y 2 de salidas pueden ser definidas la base de reglas, y 11 funciones de pertenencia pueden ser definidas para cada variable.

Pueden ser dibujadas funciones de pertenencia de forma libre o de líneas segmentadas. Usa un editor de matriz para especificar las reglas. Este editor adopta los conceptos familiares que cada celda representa una combinación de condición de entrada y permite editar dividiendo de a dos dimensiones una matriz de una alta dimensión.

Las variables ocultas son fijadas como términos difusos específicos; estos términos pueden ser cambiados con un simple click del mouse para pasar a través de diferentes divisiones de la matriz de reglas. Por un click en una celda de la regla, el usuario puede pasar a través de los diferentes términos difusos de salida y fijar la conclusión de salida deseada en la celda. Una de las características más potentes de FKB es la capacidad de mostrar el contenido de cada celda en forma gráfica. Cada término de salida difuso puede ser asociado con una gráfica; además de mostrar el nombre del término difuso en cada celda, la gráfica puede ser mostrada para permitir una mayor percepción de la topología de la matriz de reglas. Un ejemplo sería asociar la forma de un óvalo a la gráfica, entonces un óvalo será mostrado en cada celda, con el incremento del óvalo igual que como se incrementen los valores de salida.

Al ver la matriz de reglas con estas gráficas inmediatamente se generara un imagen mental de la superficie de decisión. Las gráficas tienen asociados colores, varios tipos de formas y flechas direccionales.

Después de llenar la matriz de reglas, el comportamiento de la base de reglas puede ser probado por una ventana de entrada del usuario. El usuario puede fijar los valores de entrada por barras de desplazamiento; el resultado de los valores de salida puede ser mostrado numéricamente o como la posición de una barra.

El FKB tiene un algoritmo estándar con el método de defuzzificación del centroide para la evaluación de toda la base de reglas. Además puede mostrar ploteos tridimensionales de la

superficie de decisión o como un contorno de dos dimensiones. Para un matriz de reglas de una alta dimensión puede ser un proceso tedioso llenar todas las celdas.

FUZZ-C (V. 1.0)

Fuzz-c es una herramienta de Byte Craft basada en DOS. Es un generador de código que traduce un archivo descripción de una base de reglas en una codificación en línea de código fuente C. Fuzz-c no solamente permite funciones difusas que son llamadas por un programa en C, sino también permite funciones en C que sean llamadas dentro de una regla difusa. Es mucho más barato que otras herramientas difusas, Fuzz-c permite de una forma fácil adicionar a un programa en C la capacidad de procesamiento difuso.

El usuario no necesita interactuar con Fuzz-c. Cualquier editor de texto puede ser usado para crear un archivo descripción de reglas; Fuzz-c es entonces llamado desde la línea de comandos del DOS para trasladar este archivo. En el archivo descripción de la base de reglas, una variable de entrada se define usando la sintaxis que podemos ver a continuación:

```
LINGUISTIC Temperatura TYPE int MIN 30 MAX 100 {
    MEMBER Tibio {60, 70, 80}
    MEMBER Caliente{70, 80, 100, 100}
}
```

La palabra reservada **LINGUISTIC** indica que esta es una variable de entrada. Aquí podemos definir una variable llamada **Temperatura** con dos funciones de pertenencia asociadas. Una función de pertenencia está definida por un conjunto de puntos. Tres y cuatro puntos definen funciones de pertenencia triangulares y trapezoidales respectivamente. Estos puntos también pueden ser remplazados por una función en C del usuario que permite implementar formas más complejas.

Una variable de salida se define con una sintaxis similar. Por ejemplo:

```
CONSEQUENCE Velocidad_Ventilador TYPE int MIN 0 MAX 1000 DEFUZZ CG {
    MEMBER Baja {200}
    MEMBER Alta {1000}
    ACTION { print_me(Velocidad_Ventilador); }
}
```

Aquí se ha definido un variable de salida llamada **Velocidad_Ventilador** con dos funciones de pertenencia. En este caso las funciones de pertenencia son singletons.

La declaración **DEFUZZ CG** le dice a **Fuzz_C** que use en centro de gravedad como método de defuzzificación para calcular los valores de las variables de salida.

Se puede seleccionar uno de cinco posibles métodos de defuzzificación: centro de gravedad, izquierda-mayor máximo, medio máximo, derecha-mayor máximo, y el promedio de máximos. Alternativamente, el usuario puede definir métodos de fuzzificación.

La palabra reservada **ACTION** le indica a Fuzz-C que evalúe la expresión dada cuando el valor de la salida sea calculado.

Aquí se llamará una función en C del usuario para mostrar el valor de **Velocidad_Ventilador**.

Después que las variables de entrada y salida y sus funciones de pertenencia asociadas han sido definidas, se puede especificar el bloque de reglas. La sintaxis para especificar las reglas es la siguiente:

```

FUZZY Obtener_Velocidad_Ventilador {
    IF Temperatura IS Caliente AND DeltaTemp IS Incremento
        THEN Velocidad_Ventilador IS Alta
    IF Temperatura IS Tibio OR Esmañana()
        THEN Velocidad_Ventilador IS Baja
}

```

La palabra reservada **FUZZY** identifica un bloque de reglas. El nombre dado al bloque es **Obtener_Velocidad_Ventilador**, siendo el nombre de la función en C que puede ser llamado para ejecutar las reglas. Nótese que la función del usuario **Esmañana()** es puesta dentro de la reglas.

Cualquier función del usuario puede ser evaluada como una condición en la premisa, con tal que la función retorne un valor de verdad válido. Esto lleva a una nueva forma de construcciones usando lógica difusa; una condición en la premisa puede ser una función generalizada de verdad y no necesariamente ser asociada con un función de pertenencia difusa.

Después de que Fuzz-C es aplicado a un archivo descripción de una base de reglas, se crea un archivo fuente en C que contiene varias definiciones de funciones. La función de inferencia tiene el mismo nombre de 1 bloque de reglas como por ejemplo `Obtener_Velocidad_Ventilador()`. Esta función no toma cualquier argumento, ni retorna cualquier valor, debido a que todas las variables de entradas y salida son pasadas como variables globales. Veamos la secuencia de llamado para función de inferencia:

```
Temperatura = input1; /* Asigna a Temperatura un nuevo valor */
DeltaTemp = input2; /* Asigna a DeltaTemp un nuevo valor */
Obtener_Velocidad_Ventilador(); /* ejecuta las reglas */
output1 = Velocidad_Ventilador; /* Asigna a la salida el valor de Velocidad_Ventilador.*/
```

El código C generado por Fuzz-C es fácil de leer. Trazos de las funciones de pertenencia pueden ser incluidos en el código en C como comentarios.

Fuzz-C es una excelente herramienta que permite muy fácilmente hacer construcciones de lógica difusa aplicando lenguaje C.

Fuzzy Logic ToolBox para MatLab

Este software es tratado ampliamente en este capítulo ya que constituye la base de las prácticas de simulación del curso.

Fuzzy Logic Toolbox. Versión 1.0, 1-19-95: el Toolbox es una colección de funciones construidas en el entorno de cálculo numérico de MatLab. Es un producto de Math Works Inc. que trabaja bajo el ambiente Windows. Esta construido para permitir la creación y edición de sistemas de inferencia difusa con la estructura de MatLab, o si se prefiere se puede integrar el sistema difuso dentro del ambiente gráfico (bloques) de simulación de Simulink. También permite la construcción de programas en C que pueden llamar un sistemas difuso construido con MatLab.

El Fuzzy Logic ToolBox cuenta con una interface gráfica de usuario (GUI) que sirve de herramienta para en la construcción de los sistemas, se puede de igual manera usar la línea de comandos de MatLab si se prefiere.

El Toolbox tiene tres categorías de herramientas:

- ☐ Funciones de la línea de comandos
- ☐ Herramientas iterativas gráfica (GUI)
- ☐ Bloques en Simulink y ejemplos

La primera categoría de herramientas son funciones que pueden ser llamadas desde la línea de comandos de Matlab. Muchas de estas funciones son archivos -M de Matlab que implementan algoritmos difusos especializados, los cuales pueden ser vistos digitando:

type nombre_de_la_funcion

Estos archivos pueden ser modificados, copiando y renombrando el archivo -M, con lo que se pueden ampliar el Toolbox. Esta característica es propia de MatLab, lo cual permite personalizar el entorno de trabajo con funciones de usuario.

La segunda ayuda que provee el Toolbox son unas herramientas gráficas (GUI) que no son más que interfaces gráficas para la ejecución de las funciones en la creación de sistemas de inferencia difusos, su análisis, e implementación.

La tercera categoría es un conjunto de bloques para usar con el software de simulación Simulink. El cual está diseñado para una alta velocidad en la inferencia difusa dentro del entorno de Simulink. Para instalar el Toolbox se debe haber instalado previamente Matlab y luego se instala el Toolbox como una nueva herramienta igual que otros Toolboxes de Matlab, como se hace con el de control por ejemplo.

El tutorial del Toolbox es de gran ayuda para usuarios inexpertos. Debido a que este los introduce con ejemplos didácticos a la lógica difusa, para posteriormente guiarlo en el diseño de sistemas difusos. Cubre preguntas como: Qué es la lógica difusa?, Por qué usar lógica difusa?. También muestra un ejemplo que es muy común en la enseñanza de la lógica difusa el cual consiste en comparar la toma de decisiones aplicando lógica difusa versus la lógica tradicional en este caso se refieren al ejemplo del pago de propina por la calidad del servicio obtenido. Después el tutorial da los fundamentos de la lógica difusa: Conjuntos difusos, tipos de funciones de pertenencia, operadores lógicos, reglas if-then, sistema de inferencia difuso, fuzzificación y defuzzificación, todos ellos induciendo al uso práctico del Toolbox y acompañados de ejemplos como: el conjunto difuso de los días que

comprenden el fin de semana , Los meses y las estaciones del año, y el típico ejemplo de la estatura de las personas.

Como se mencionó anteriormente la Toolbox tiene herramientas GUI para la construcción, edición y observación de sistemas de inferencia difusa que son:

Editores

- ▣ **FIS Editor (Fuzzy Inference Systems):** Editor del sistema de inferencia
- ▣ **Rule Editor:** Editor de la reglas if-then
- ▣ **Membership Function Editor:** Editor de funciones de pertenencia

Visualizadores (herramientas de sólo lectura)

- ▣ **Rule Viewer:** Visualizador de activación de reglas
- ▣ **Surface Viewer:** Visualizador de la superficie de decisión o de inferencia

La figura 7-5 muestra los cinco GUT's utilizados en la Toolbox Fuzzy de Matlab para la creación de un sistema de inferencia difusa.

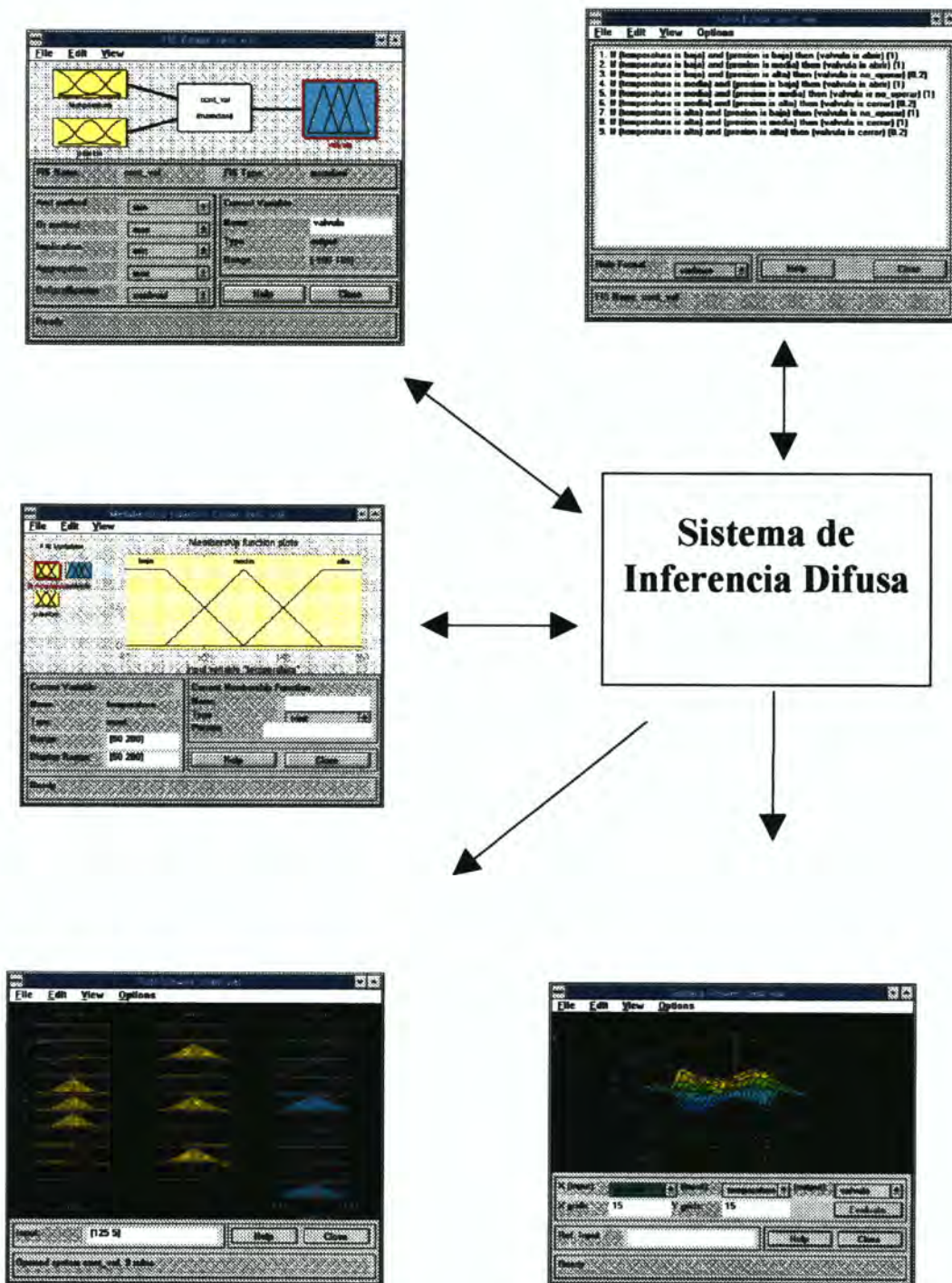


Figura 7-5 Editores GUI's

La siguiente gráfica es un guía para la construcción de sistemas difusos aplicando los editores GUI:

Caso General

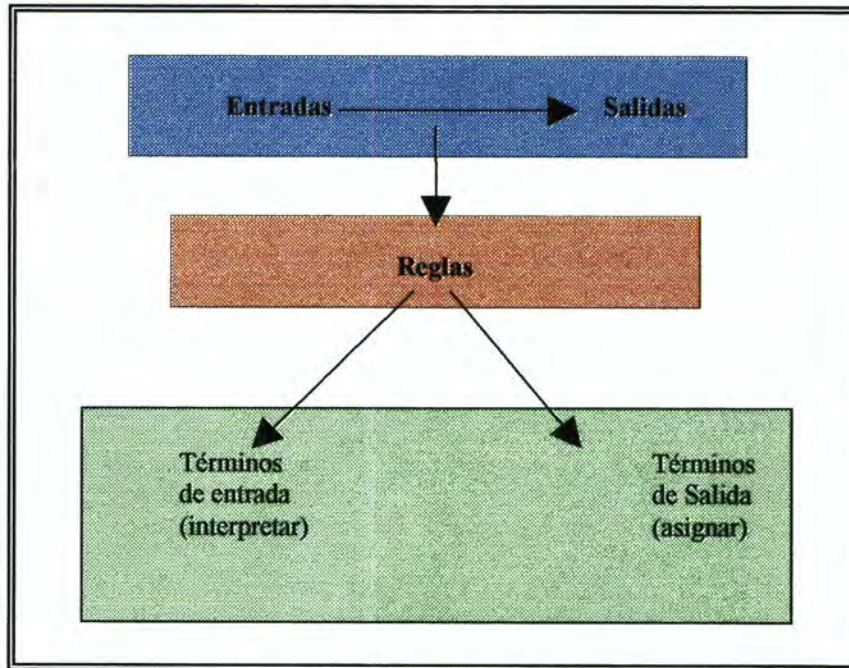


Figura 7-6 Modelo de un caso general

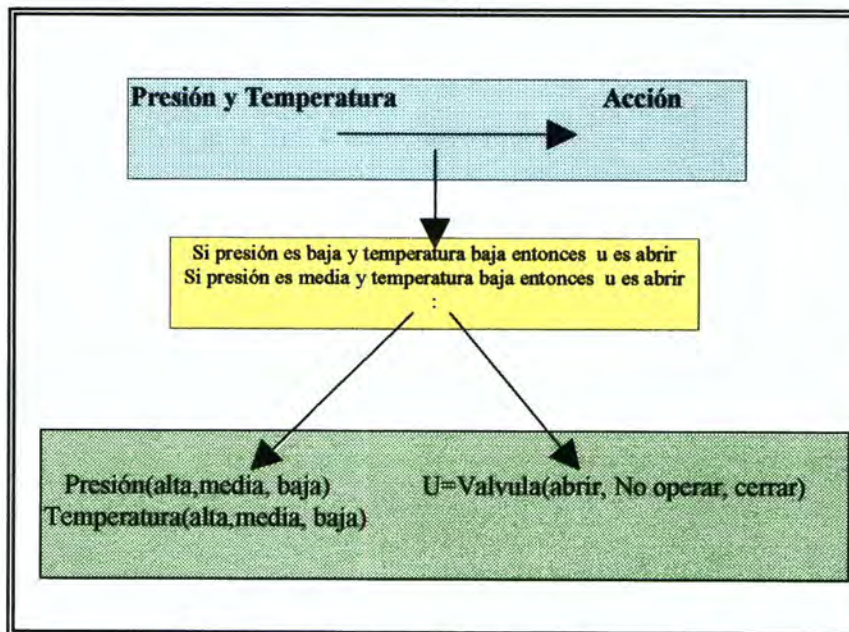


Figura 7-7 Modelo de un caso específico

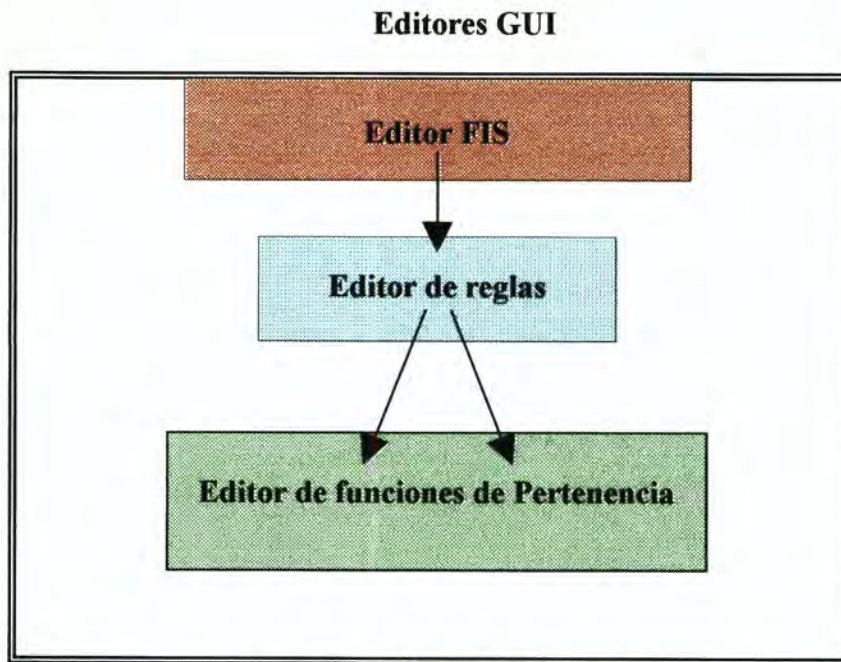


Figura 7-8 Uso de editores GUI's

Para cargar el editor FIS se debe digitar fuzzy en la línea de comandos de Matlab. Los archivos que genera el FIS editor son de extensión .FIS, si se desea cargar un archivo que se está trabajando se puede digitar fuzzy nombre_del_archivo, no es necesaria la extensión. Si se desea se puede digitar fuzzy tipper que carga un archivo ejemplo que trae el Toolbox de un sistema de inferencia difusa para el pago de propinas.

El FIS editor brinda información general sobre el sistema de inferencia difusa. En la parte superior aparece el nombre del archivo y el tipo de inferencia usado, por defecto es Mamdani. Aunque puede ser usado el tipo Sugeno (ver capítulo 3 clasificación de sistemas difusos).

El sistema por defecto en el FIS Editor es SISO (simple entrada simple salida) pero pueden ser adicionadas más variables de entrada.

A manera de práctica se describirá el diseño de un sistema para controlar una válvula a partir de la información dada por un sensor de presión y uno de temperatura. Se aclara que el objetivo es dar una introducción al manejo de la herramienta como tal.

Primero llamemos el Toolbox digitando fuzzy desde Matlab, lo cual cargará el editor FIS como se ve en la figura 7-9.

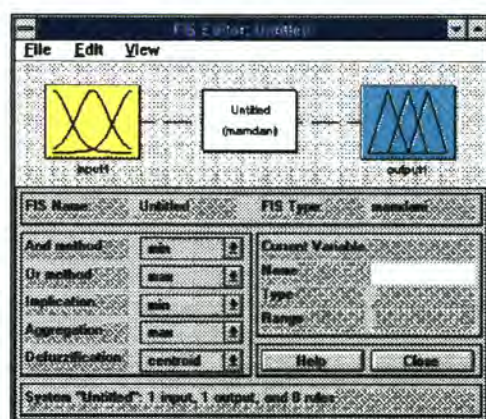


Figura 7-9 Editor FIS

A continuación describiremos los menús del editor:

File: usado para grabar o dar un nombre al sistema a implementar, se llama seleccionando la opción File con el mouse o con las teclas Alt+F; se pueden ver 9 opciones en este menú: las dos primeras permiten crear nuevos sistemas de tipo Mamdani o Sugeno como una nueva ventana de trabajo. Las tres siguientes permiten: abrir un archivo, grabar, y renombrar archivos .fis. Las tres opciones restantes permiten grabar el sistema en el espacio de trabajo de Matlab (workspace) como una variable que contiene toda la información del sistema, editar una de estas variables o renombrarla. La ultima opción cierra la ventana.

Edit: permite en su orden; adicionar y remover variables de entrada y de salida, y una opción de deshacer (Undo) útil cuando nos arrepentimos de haber efectuado una operación previa.

Por defecto los nombres dados a las variables de entrada son input1, input2,... y salida output1, output2.... Cada nueva variable de entrada adiciona una representación en forma de una caja amarilla al sistema en el editor, igual sucede con las salidas pero de color azul. Para dar un nombre a las nuevas variables de entrada o salida, se debe seleccionar la variable con el mouse, la variable seleccionada colocará un contorno rojo la caja que la representa. Una vez seleccionada la variable, en un recuadro blanco, lado derecho de la pantalla aparecerá el nombre actual de la variable, digite el nuevo nombre dentro de este recuadro, verá como se actualiza la ventana con el nuevo nombre. Debajo del recuadro aparece el tipo de variable (input, output) y su rango.

Ingresemos para el ejemplo las variables de entrada Temperatura y Presión, y para la salida la variable Válvula usando la opción respectiva en Edit .

Se aclara que las opciones de Edit varían al cambiar de editor (recuerde existen tres editores FIS, de reglas y de funciones de pertenencia). Las cajas que representan las variables de entradas están relacionadas con las de salida por una caja blanca que contiene el nombre del sistema y el tipo de inferencia.

El menú View: de sus cinco opciones sólo están activas 4 ya que la primera no tiene ningún sentido porque es para ir al editor FIS. Las dos siguientes permiten pasar del editor FIS al

editor de reglas o al de funciones de pertenencia. De igual manera funciona en los otros dos editores y los dos visualizadores.

En el lado izquierdo de la ventana se puede escoger el operador para los métodos And, Or, Implicación, agregación y el método de defuzzificación, seleccionando la barra de cada botón se podrá cambiar el método. También existe una barra para cerrar la ventana y obtener ayuda. El editor tiene todas las opciones de ventana del ambiente Windows. Para el ejemplo se dejarán los parámetros mostrados por defecto.

Para pasar al editor de funciones de pertenencia se puede dar doble click directamente sobre una variable o desde la opción del menú View.

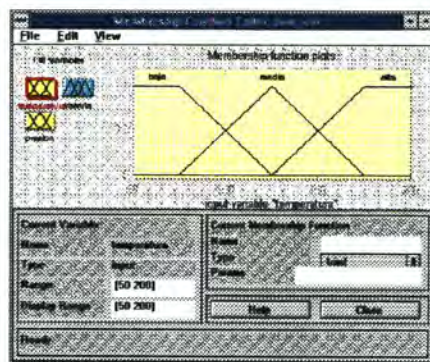


Figura No 7-10 Editor GUI de funciones de pertenencia

Este editor difiere del FIS en el menú Edit, ya que se cambia a las opciones de adicionar una función de pertenencia, adicionar una función de pertenencia habitual y remover una o todas las funciones de pertenencia. La función de pertenencia habitual muestra un recuadro en el cual aparece el archivo del (M) que la contiene y su nombre, si se desea esta puede ser cambiada.

Si se adiciona una MF (función de pertenencia) al sistema, el Toolbox mostrará una ventana que permite escoger el tipo de MF de 11 posibles e ingresar al mismo tiempo varias de éstas (de 1 a 9). Veamos los once tipos:

- `dsigmf` - Diferencia de dos sigmoid MF's.
- `gauss2mf` - Curva de dos MF gaussianas.
- `gaussmf` - Curva Gaussiana.
- `gbellmf` - Generalizada curva de campana.
- `pimf` - Curva Forma Pi.
- `psigmf` - Producto de dos sigmoid MF's.
- `smf` - Curva de forma S.
- `sigmf` - Curva Sigmoide.
- `trapmf` - Curva Trapezoidal.
- `trimf` - Curva Triangular.
- `zmf` - Curva de forma Z.

La forma triangular está compuesta por 3 puntos en forma de triángulo, la Trapezoidal es una triangular truncada y tiene 4 puntos, las gaussianas requieren de dos parámetros de la curva, la curva de campana generalizada requiere tres parámetros, la sigmoideal se divide en una `sigmf` sencilla y dos compuestas `dsigmf` y `psigmf`, las demás curvas son la S, la Z y las curvas Pi.

Las funciones serán distribuidas uniformemente en el rango de la variable. Para ser editadas se seleccionan con el mouse, en editor se aprecian las variables al lado izquierdo de la ventana, al lado derecho un recuadro muestra la variable seleccionada para adicionar o modificar sus funciones de pertenencia, en la casilla del lado derecho (Name) se digita el

término o etiqueta para la función de pertenencia, en (Type) el tipo de función de las once disponibles, en (Params) los puntos que caracterizan los vértices o parámetros de las diferentes funciones. Las etiquetas de las MF deben ser cadenas de caracteres seguidas, usar el guión (_) si es necesario ejemplo no_operar.

Para el ejemplo se escogerán las siguientes funciones de pertenencia, seleccione cada una de las variables de entrada y salida digitadas previamente (Temperatura, Presión y Válvula) y de los siguientes valores:

Entrada 1

'temperatura'

Range=[50 200]

Número de Mfs 3

MF1 = ' baja': 'trapmf', [50 50 75 125] (función Trapezoidal)

MF2 = ' media': 'trimf', [75 125 175] (función triangular)

MF3 = ' alta': 'trapmf', [125 175 200 200] (función Trapezoidal)

Entrada 2

'presión'

Range=[0 10]

Número de Mfs 3

MF1 = ' media': 'trimf', [1 5 9] (función triangular)

MF2 = ' baja': 'trapmf', [0 0 1 4] (función Trapezoidal)

MF3 = ' alta': 'trapmf', [6 9 10 10] (función Trapezoidal)

Salida 1 (sólo existe una salida)

'válvula'

Range=[-100 100]

Número de Mfs 3

MF1='cerrar':trimf,[-100 -100 0] (función triangular)

MF2='abrir':trimf,[0 100 100] (función triangular)

MF3='no_operar':trimf,[-100 0 100] (función triangular)

En el lado inferior izquierdo aparece el nombre de la variable su tipo y el rango que se quiere dar y el de visualización. Las opciones File y View funcionan igual que en el FIS editor.

Cambiemos al editor de reglas seleccionando esta opción desde el menú View.



Figura No 7-11 Editor GUI de Reglas

Editor de Reglas: en el editor de reglas se pueden ingresar las reglas de tres maneras: verbal, simbólicamente e indizada, que son seleccionadas desde la parte inferior de la ventana de un menú de barra. En el menú Edit se encuentra una opción para deshacer una ejecución previa. También existe un menú (options) para definir el idioma cuando se edita la regla de manera verbal (inglés, francés y alemán).

Para ingresar la regla se digita esta en el formato if-then y a continuación Ctrl+Enter

Las reglas pueden digitarse de la siguiente manera:

if temperatura is baja and presión is baja then válvula is abrir

una vez se termina se da Ctrl + Enter y el Toolbox le mostrará el siguiente formato:

1. If (temperatura is baja) and (presión is baja) then (válvula is abrir) (1)

El número 1, colocado por editor indica el número de la regla, y el número (1) es el peso por defecto de la respectiva regla pero puede ser variado a un valor entre 0 y 1. Si la regla tiene algún problema como una variable o una MF desconocida, la regla aparecerá con un símbolo de # de la siguiente manera (para al caso de escribir is en vez de es):

if temperatura es baja and presión is baja then válvula is abrir

veamos la base de reglas para nuestro ejemplo usando los tres formatos:

Formato simbólico (symbolic)

1. (temperatura==baja) & (presion==baja) => (valvula=abrir) (1)
2. (temperatura==baja) & (presion==media) => (valvula=abrir) (1)
3. (temperatura==baja) & (presion==alta) => (valvula=no_operar) (0.2)
4. (temperatura==media) & (presion==baja) => (valvula=abrir) (1)
5. (temperatura==media) & (presion==media) => (valvula=no_operar) (1)

6. (temperatura==media) & (presion==alta) => (valvula=cerrar) (0.2)
7. (temperatura==alta) & (presion==baja) => (valvula=no_operar) (1)
8. (temperatura==alta) & (presion==media) => (valvula=cerrar) (1)
9. (temperatura==alta) & (presion==alta) => (valvula=cerrar) (0.2)

Formato Verbal ingles(verbose)

1. If (temperatura is baja) and (presión is baja) then (válvula is abrir) (1)
2. If (temperatura is baja) and (presión is media) then (válvula is abrir) (1)
3. If (temperatura is baja) and (presión is alta) then (válvula is no_operar) (0.2)
4. If (temperatura is media) and (presión is baja) then (válvula is abrir) (1)
5. If (temperatura is media) and (presión is media) then (válvula is no_operar) (1)
6. If (temperatura is media) and (presión is alta) then (válvula is cerrar) (0.2)
7. If (temperatura is alta) and (presión is baja) then (válvula is no_operar) (1)
8. If (temperatura is alta) and (presión is media) then (válvula is cerrar) (1)
9. If (temperatura is alta) and (presión is alta) then (válvula is cerrar) (0.2)

Formato Verbal frances(verbose)

1. Si (temperatura est baja) et (presión est baja) alors (válvula est abrir) (1)
2. Si (temperatura est baja) et (presión est media) alors (válvula est abrir) (1)
3. Si (temperatura est baja) et (presión est alta) alors (válvula est no_operar) (0.2)
4. Si (temperatura est media) et (presión est baja) alors (válvula est abrir) (1)
5. Si (temperatura est media) et (presión est media) alors (válvula est no_operar) (1)
6. Si (temperatura est media) et (presión est alta) alors (válvula est cerrar) (0.2)
7. Si (temperatura est alta) et (presión est baja) alors (válvula est no_operar) (1)
8. Si (temperatura est alta) et (presión est media) alors (válvula est cerrar) (1)
9. Si (temperatura est alta) et (presión est alta) alors (válvula est cerrar) (0.2)

Formato indexado.

- 1 2, 2 (1) : 1
- 1 1, 2 (1) : 1
- 1 3, 3 (0.2) : 1
- 2 2, 2 (1) : 1
- 2 1, 3 (1) : 1
- 2 3, 1 (0.2) : 1

3 2, 3 (1) : 1
 3 1, 1 (1) : 1
 3 3, 1 (0.2) : 1

Para entender este último debemos tener en cuenta que el Toolbox establece un orden para las variables de entrada igual que para las etiquetas o MF.

En este ejemplo las dos primeras columnas corresponden a los estados de la variable temperatura y presión para cada regla respectivamente: las tres primeras reglas del ejemplo tienen 1 al principio lo que corresponde a temperatura baja después de la coma se encuentran las etiquetas de las variables de salida para este caso es una sola variable (válvula) y en las dos primeras reglas corresponden a la etiqueta abrir de la variable válvula. La columna de los paréntesis corresponde a los pesos de las reglas y el valor después de los dos punto indica si la regla es de tipo OR "2" o AND "1", para una mayor comprensión compare los formatos verbal e indexado.

Pasemos al visualizador de reglas usando la opción del menú View. En el visualizador de reglas podemos ver el nivel de activación de las reglas de manera gráfica, regla por regla de manera paralela variando con el mouse el valor de las variables de entrada o digitando los valores en la casilla input como un vector de los valores de las entradas. En la parte inferior de la ventana se indica el número de reglas. También se puede observar los valores crisp de la variable de salida al lado derecho, donde se ve reflejado dicho valor como la suma de la activaciones individuales de las reglas.

Para el ejemplo podremos ver qué valor debe ser dado a la válvula para el control de presión y temperatura.

Cambiamos al visualizador de superficie desde el menú de View. En el visualizador de superficie podemos ver el mapeo de las variables de entrada a la variables de salida en dos dimensiones o en tres para dos entradas, simplemente escogiendo las variables a visualizar en la casilla apropiada. Se puede modificar la grilla en ambos ejes (x,y) también se puede cambiar la variable de salida cuando se tienen más de una de estas en el sistema. Con esta ultima opción terminamos el ejemplo.

Otras características: los archivos **.fis** son archivos textos, un sistema puede ser grabado como una matriz cuando se graba como una variable en el espacio de trabajo o como un archivo **.fis** ambos representarán el mismo sistema pero sus formatos son diferentes.

Desde la línea de comandos de Matlab se pueden utilizar las funciones del Toolbox como el max, min, etc. En el caso de dos matrices el operador min devolverá el mínimo evaluando elemento a elemento la matrices. Ejemplo:

```
a = [2 2; 3 4;]
```

```
b = [1 2; 5 9;]
```

```
min(a,b)
```

```
ans =
```

```
1    2
```

```
3    4
```

Si deseamos trabajar en la línea de comandos podemos cargar un archivo **.fis** en una variable de la siguiente manera:

```
a = readfis('nombre_archivo.fis');
```

en *a* se guardará una matriz con toda la información del sistema en forma numérica. Para ver el contenido de un archivo .fis podemos digitar :

```
type nombre_archivo.fis
```

getfis(a) mostrará información genérica del sistema cargado en la matriz *a*

```
getfis(a)
    Name = Con_VAL
    Type = Mamdani
    NumInputs = 2
    InLabels =
        temperatura
        presión
    NumOutputs = 1
    OutLabels =
        válvula
    NumRules = 9
    AndMethod = min
    OrMethod = max
    ImpMethod = min
    AggMethod = max
    DefuzzMethod = centroid
```

con **getfis(a,'name')** muestra el nombre

getfis(a,'input',1) muestra los datos de la variable de entrada 1

getfis(a,'output',1) muestra los datos de la variable de salida 1

getfis(a,'output',1,'mf',1) muestra la función de pertenencia 1 de la variable de salida 1

`setfis` permite hacer cambios en un FIS

```
a=setfis(a,'name','Con_VAL')
```

aquí se le cambió el nombre al sistema.

La función `show(a)` muestra una lista numerada del sistema útil para depuración de éste.

Los GUI's pueden ser llamados desde la línea de comandos de la siguiente manera:

`fuzzy(a)` carga el editor FIS con el sistema de la matriz a

`mfedit(a)` carga el editor de MF

`ruleedit(a)` editor de Reglas

`ruleview(a)` visualizador de inferencia

`surfview(a)` visualizador de superficie

Para mostrar el sistema de inferencia existen tres funciones: `plotfis`, `plotmf` y `gensurf`. Su formato es:

`plotfis(a)` muestra el sistema como bloques

`plotmf(a,'input',1)` muestra la gráfica de las MF de la variable de entrada 1

`gensurf(a)` muestra la superficie tridimensional de mapeos

Las funciones de la línea de comandos se resumen a continuación

`addmf` - Adiciona MF a un FIS

addrule - Adiciona reglas a un FIS.
addvar - Adiciona variables a un FIS.
defuzz - Defuzzify MF.
evalfis - Calcula el resultado de un FIS.
evalmf - Función de evaluación genérica de MF.
gensurf - Genera la superficie de salida FIS.
getfis - Trae las propiedades de in FIS.
mf2mf - Traslada parámetros entre funciones.
mfstrrch - Stretch MF.
newfis - Crea un nuevo FIS.
parsrule - Parse fuzzy rules.
plotfis - Display FIS input-output diagrama.
plotmf - Display todas las MF de una variable.
readfis - Carga un FIS de un disk.
rmmf - Remueve MF de un FIS.
rmvar - Remueve variables de un FIS.
setfis - Conjunto de propiedades.
showfis - Display annotated FIS.
showrule - Display FIS reglas.
writefis - Graba FIS a un disk.

Dentro del proceso de construir sistemas es importante evaluar de manera rápida el resultado; para ello podemos usar **evalfis** de la siguiente manera: **evalfis([1 2], a)**. Se deben tener en cuenta los rangos de las entradas:

```
evalfis([50 1], a)
```

```
ans = 67.3333
```

Para varios valores de entrada se puede usar la siguiente sintaxis:

```
evalfis([50 1; 60 3], a)
```

```
ans =
```

```
67.3333
```

```
61.6316
```

El uso de Toolbox con Simulink permite trabajar sistemas difusos creados con FIS en el entorno de simulación de Simulink, el cual trabaja a través de bloques a los cuales se les da sus respectivos parámetros para el proceso de simulación, tiene características tales como permitir la animación, bloques generadores de señales, la visualización gráfica de entradas y salidas.

El sistema difuso es insertado como un bloque de control el cual tendrá sus respectivas entradas y salidas de acuerdo con las definiciones en el FIS. Se puede traer el bloque de simulación digitando fuzblock de la línea de comandos que mostrará el bloque que se ve a continuación

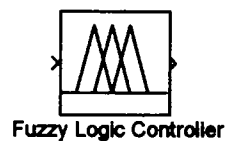


Figura 7-12 Bloque de Control difuso para Simulink

Para ver un ejemplo se puede digitar sltank desde la línea de comandos de Matlab el cual traerá la simulación del control de nivel de un tanque como se ve en la figura 7-13.

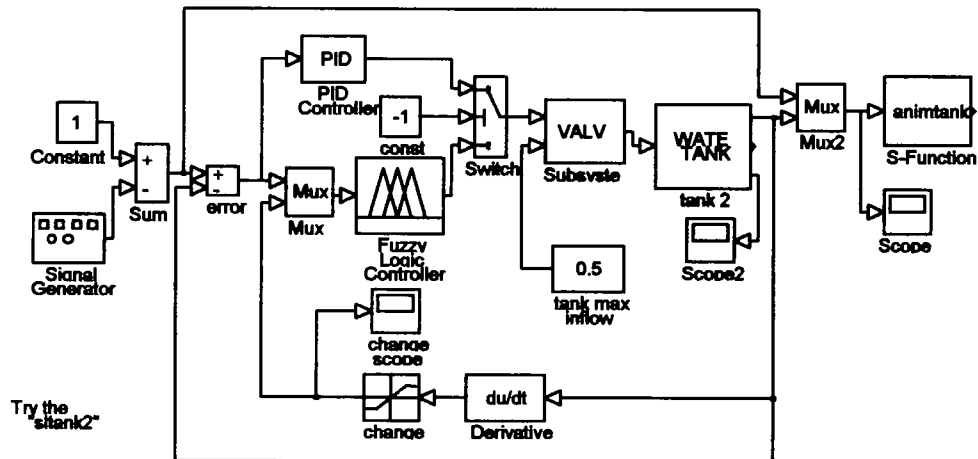


Figura 7-13 Simulación del control de nivel de un tanque Bloque (Simulink)

El Toolbox provee también un algoritmo adaptativo-neuro-difuso llamado ANFIS, que es básicamente un algoritmo backpropagation, que basado en datos de entrada y salida logra la sintonía para un sistemas FIS. Está limitado a sistemas tipo Sugeno de primer orden y pesos unitarios para las reglas. Posee dos librerías en C para la generación de sistemas difusos estas son: fisman.c y fis.c. Cuenta además con varios ejemplos de aplicación con animación: control de una bola en equilibrio (Ball juggling) el cual puede ser visto digitando juggler, robot de dos brazos unidos el cual usa el ANFIS, un vehículo con tracción delantera que trabaja en Simulink, digite sltbu para cargarlo. El péndulo invertido también trabaja en Simulink cárguelo digitando slcp.

Clasificación en la literatura encontramos la clasificación que hace Pedrycz [PW95] de los recursos de software en tres grupos como muestra la tabla 7-2, que recoge todas las categorías de software usadas para el diseño y simulación de sistemas difusos.

Tabla 7-2 Clasificación de Software

| Clase de software | Característica |
|--|---|
| 1. Plataformas especializadas para diseño de sistemas difusos | <ul style="list-style-type: none"> • Permiten el diseño de un controlador difuso con un mínimo esfuerzo de programación. • Poseen interfaces que los hacen amistosos para el usuario • Prácticamente orientados a la validación estática |
| 2. Herramientas generales de simulación | <ul style="list-style-type: none"> • Permiten bastante flexibilidad por que cortan una cantidad excesiva de un tedioso y detallado trabajo de la programación. • El diseño se reduce a la manipulación de bloques gráficos del software de simulación |
| 3. Lenguajes de programación estándar o de alto nivel (compiladores o intérpretes) | <ul style="list-style-type: none"> • Lenguajes como el C, Pascal, VisualBasic, Prolog etc. • Su desventaja surge en la cantidad de código necesario para cubrir todos los detalles del controlador |

8. DISPOSITIVOS DE HARDWARE PARA LA IMPLEMENTACIÓN DE CONTROLADORES DIFUSOS

Para la implementación de los sistemas difusos actualmente se encuentran diferentes plataformas de hardware como por ejemplo los procesadores difusos dedicados, microprocesadores y microcontroladores de uso general y controladores lógicos programables: PLC's.

Cuando un microcontrolador o un microprocesador convencional son usados en conjunto con un algoritmo bajo algún software de lógica difusa, las reglas que constituyen la base del algoritmo son evaluadas en secuencia una después de otra. Una vez todas las reglas sean evaluadas, sus salidas son combinadas para dar un simple valor que será defuzzificado.

En cambio si se usa un procesador difuso dedicado, las reglas son evaluadas en paralelo. El método de procesamiento en paralelo sugiere un rápido ciclo de procesamiento. Sin embargo en este caso, la adquisición de datos y la salida tienen que ser realizados usando periféricos convencionales. El tiempo ganado en el procesamiento de reglas en paralelo se puede perder en la adquisición de los datos mediante los periféricos externos.

Para muchos sistemas de control, la mejor solución es usar un algoritmo difuso sobre un microcontrolador o con unos periféricos internos rápidos. En estos casos, el esquema

secuencial del procesamiento de las reglas es transparente para el usuario y el proceso parece haber sido realizado en paralelo.

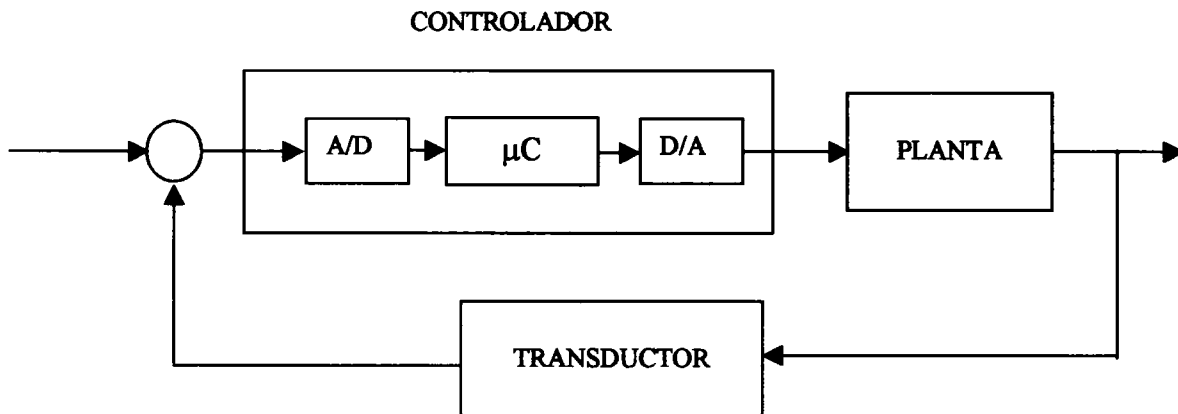


Figura No 8-1 Aplicación del microcontrolador en un lazo control

Algunos fabricantes han integrado un coprocesador difuso sobre los microcontroladores convencionales permitiendo reunir en un solo chip la funcionalidad de las dos unidades.

Algunos de las soluciones de hardware disponibles para la implementación de los sistemas difusos son:

Coprocesador difuso C991 (81C991): Es un controlador difuso de propósito general o clasificador difuso para aplicaciones en tiempo real. Muestra gran flexibilidad en tareas que requieren de un amplio rango de control y de clasificación y que demandan procesamiento en tiempo real y gran resolución. El C991 soporta 10 modos de operación diferentes, 8 operadores de inferencia diferentes y 4 métodos de defuzzificación. Su código se programa mediante el software FLDE.

Familia de microcontroladores MCS-96: están equipados con periféricos internos de gran desempeño que permiten un manejo fácil de la adquisición de datos, el acondicionamiento de señales y de las salidas. Tales periféricos tienen un amplio rango de direccionamiento y el poderoso conjunto de instrucciones que la familia ofrece los hace muy útiles para las aplicaciones de lógica difusa.

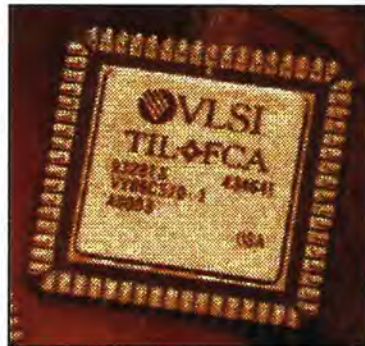


Figura 8-2 Procesador difuso VY86C570

Beneficios y características del VY86C570:

- ☐ Bajo consumo de potencia
- ☐ Gran desempeño (opera a 20 MHz)
- ☐ Alta resolución
- ☐ Extremadamente rápido (velocidad de inferencia de 70 microsegundos para evaluar 800,000 reglas por segundo)
- ☐ Hasta 1024 entradas o salidas (dependiendo de la cantidad de memoria implementada con la base de reglas).

Controlador difuso reconfigurable Fuzzy Set 001

Descripción:

- ☐ Tiene dos entradas y una salida.
- ☐ El motor de inferencia implementa relaciones tipo Mamdani min-max.
- ☐ El bloque de defuzzificación usa el método del Centro de Gravedad.
- ☐ Cada entrada o salida consta de 5 términos lingüísticos: NL, NS, Z, PS, PL.
- ☐ Implementa 13 reglas fuzzy.
- ☐ Los pines de control activan el reorganización del conjunto de reglas.
- ☐ Requiere una fuente de voltaje dual de +6V y -6V.
- ☐ Tiene una alta velocidad por encima de 6 MFLIPS;
- ☐ Encapsulado de 48-Pin tipo DIL.

Coprocesador difuso Fuzzy Set-002

Descripción

- ☐ Diseñado para aplicaciones en tiempo real y sistemas de alta velocidad.
- ☐ Es un co-procesador difuso diseñado con tecnología VLSI.
- ☐ Diseñado con técnicas análogas y digitales, la parte digital se usa para la configuración del chip, mientras que la parte difusa para los cálculos difusos.
- ☐ El motor de inferencia implementa relaciones tipo Mamdani min-max.
- ☐ El bloque de defuzzificación usa el método del Centro de Gravedad.

Controlador difuso AL220:

Fabricado por Adaptive Logic, este microcontrolador dedicado presenta las siguientes características:

- ▣ 4 entradas y salidas analógicas
- ▣ Chip análogo programable (PAIC(tm)).
- ▣ Conversores A/D y D/A internos.

Elimina el uso de look up tables mediante una función de pertenencia flotante.

Aplicaciones:

- ▣ Control de movimiento y posición
- ▣ Control de temperatura
- ▣ Manejo de potencia
- ▣ Sensores inteligentes
- ▣ Control Tipo PID
- ▣ Control de motores
- ▣ PLD Análogo

PLC Difuso:

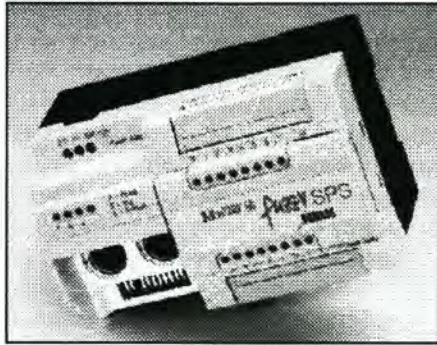


Figura 8-3 PLC difuso

El PLC difuso es programado mediante una versión ampliada del software para desarrollo de sistemas de lógica difusa FuzzyTech de Inform Software. Este software se ejecuta sobre un PC y es enlazado al PLC difuso mediante el cable serial RS 232 o por medio del bus RS 485. El PLC difuso tiene la característica de permitir un depuramiento On line, de esta manera cualquier modificación se puede trasladar al sistema sin parar la operación.

Como se ha visto hasta ahora los controladores de lógica difusa pueden ser implementados con microprocesadores de propósito general. Esta técnica es muy popular debido a que la lógica difusa no demanda grandes cálculos o gran capacidad de memoria. El código utilizado por estos controladores requiere de menos de 1000 bytes de almacenamiento en memoria, inclusive para la mayoría de aplicaciones, un microprocesador de 8 bits puede manejar los cálculos necesarios.

El microprocesador es usado luego de que el modelo ha sido verificado mediante simulación a través de alguna herramienta de software (ver capítulo 7). El código resultante que compone el controlador difuso, es grabado en una memoria ROM.

Son usados circuitos integrados dedicados difusos en situaciones en donde se requiera una mayor velocidad o la evaluación de una gran cantidad de reglas. Los chips difusos son de 10 a 100 veces más rápidos que los microprocesadores de propósito general que corren bajo un código de software de lógica difusa.

Togai Infralogic Inc. Por ejemplo, fabrica toda una línea de controladores difusos dedicados que, según la compañía son 10 veces más rápidos que un 486 de Intel configurado como un controlador difuso. Su RISC (Reduced Instruction-Set Computer) **FC110 IC** maneja hasta 800 reglas y las procesa a una velocidad de 200000 reglas por segundo.

El controlador difuso de **Siemens**, el **81C99** es un coprocesador que sirve de interface para la mayoría de los microprocesadores. Actúa como un acelerador para los microprocesadores de propósito general. Es capaz de procesar las reglas a una velocidad de 7.9 millones de reglas por segundo.

Adaptive Logic (anteriormente **American Neuralogix Inc.**) También fabrica un controlador dedicado difuso, el **NLX 220**. Maneja un máximo de 4 entradas y salidas análogas. Realiza la conversión de entradas/salidas en tiempo real a través de un chip con conversores A/D y D/A. Soporta hasta 111 reglas y las procesa a una velocidad de 800000 reglas por segundo. Viene acompañado además, con un sistema de desarrollo, permitiendo a los usuarios simular y configurar el **NLX 220** rápidamente para producir nuevos productos.

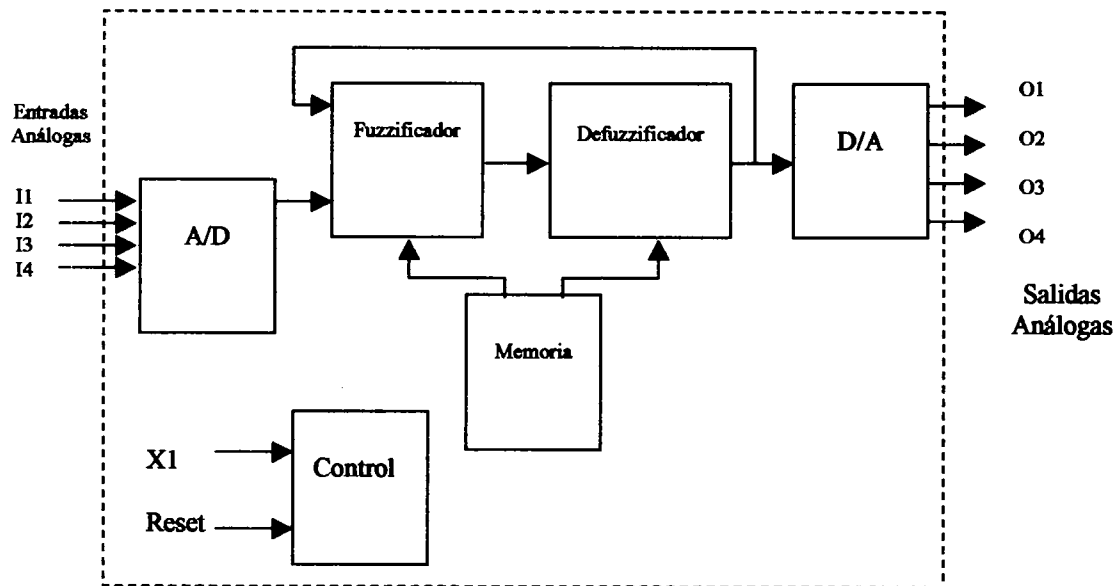


Figura 8-4 Arquitectura interna NLX 220.

Fuzzy Logic Embedded Controller es un completo sistema de desarrollo ofrecido por Hitachi que permite implementar sistemas difusos expertos con los microcontroladores H8/300, H8/300L, H8/300H, y SH. Este paquete incluye dos productos de software difuso el TILShell TM 3.0 (ver capítulo 7) y el MicroFPL.

Algunas características de este sistema de desarrollo son:

- Su eficiente código hace que se requiera menos memoria para implementar el sistema y que sea transportable a diferentes programas.

- Permite crear múltiples métodos de defuzzificación, funciones de pertenencia simples y complejas para el compilador Fuzzy-C.
- Reduce en 2/3 el tiempo de desarrollo comparado con otros métodos convencionales
- Es muy fácil de usar y aprender y no requiere experiencia previa con lógica difusa para su uso.

Para conseguir este paquete se puede llamar al 1-800-285-1601, ext. 20 o en el sitio web:

<http://www.hitachi.com>

Control difuso basado en el 8XC196 Resulta razonable utilizar el microcontrolador 8XC196 en un lazo de control difuso debido a su bajo costo, pequeño tamaño, manejo de potencia, y altas velocidades de reloj. Entre las aplicaciones típicas del 8XC196 se incluyen: Control en lazo cerrado y procesamiento de señales como control de motores, discos duros, frenos ABS e instrumentación médica.

El poder del conjunto de instrucciones se basa en que utiliza una arquitectura registro-registro con varios modos de direccionamiento y un rico conjunto de periféricos. La adquisición de datos y el procesamiento son realizados fácil y eficientemente. El software puede ser implementado con un mínimo de memoria y rápida ejecución (por encima de 50MHz) localizando el código y los datos en la memoria interna del chip. Es viable la utilización del 8XC196 debido a que se reduce el tiempo dedicado para la creación del código.

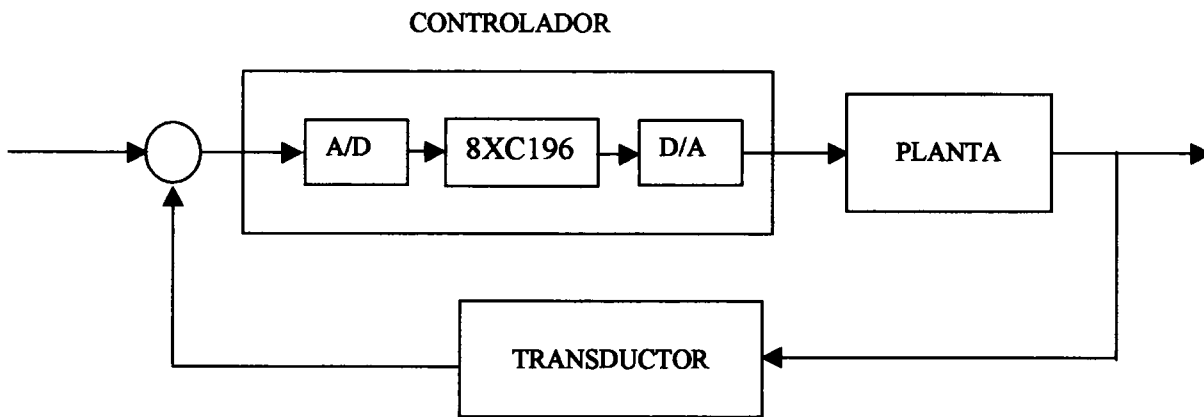


Figura No 8-5 Aplicación del microcontrolador 8XC196 en un lazo control

En el gráfico anterior podemos observar la aplicación de control difuso en un lazo cerrado de control mediante la implementación del sistema de inferencia difusa utilizando un microcontrolador 8XC196 y dos conversores análogos digitales

Podemos programar microcontroladores de la familia MCS-51 de Intel con la Herramienta **FLASH**, Fuzzy Logic Applications Software Helper, que es un generador de código que permite implementar un conjunto de subrutinas para desarrollar aplicaciones control con lógica difusa. El usuario escribe un archivo de descripción de tareas (CTDF, por sus siglas en inglés) para ser ingresado en FLASH. El archivo CTDF consta de cinco elementos: entradas (inputs), salidas (outputs), términos (terms), reglas (rules), y opciones.

No se requiere un gran conocimiento de la lógica difusa para su uso. El CTDF puede ser escrito con cualquier editor de texto (ASCII editor).

FLASH requiere recursos mínimos del computador. Viene incluido con simulador para PC

(FLASH Simulator), y un programa ensamblador (RASM51).

El FLASH Simulator genera salidas apartir de las entradas dadas al PC. El simulator puede ser usado para verificar varias entradas y hacer una sintonía fina (fine-tune) de los términos o reglas. Requiere un IBM PC o Compatible, con DOS 3.0 o superior, y un sistema mínimo con controladores de la familia MCS-51.

Motorola, ofrece dos entornos de desarrollo de lógica difusa para generar el código objeto que implemente la fuzzificación, la defuzzificación y el motor de inferencias. Este código es usado para diseñar un sistema difuso de control con sus microcontroladores.

Uno de estos entornos de desarrollo es el KBG11B.EXE (Knowledge Base Generator) que soporta un motor de inferencias difusas llamado FUZZY11B.ASM. KBG11B.EXE funciona bajo MS-DOS, y brinda una interface gráfica para la creación de las funciones de pertenencia de entrada y de salida. El otro entorno de desarrollo es el visto en el capítulo 7 Fuzzy Inference Development Environment (FIDE), de Apronix, que funciona en IBM-PC, para el entorno gráfico de Microsoft Windows. FIDE ofrece una interface gráfica para desarrollar, depurar (de-bugging), y simular los sistemas a implementar.

En la literatura se destaca una aplicación para el reconocimiento de caracteres en cheques usado en los Estados unidos OCA (Optical Character Asso-ciator), que fue desarrollado con la herramienta FIDE de Apronix. Para una amplia información de esta aplicación remítase a SEMICONDUCTOR MOTOROLA APPLICATION NOTE "Optical Character Recognition Using Fuzzy Logic" de William A. Gowan..

9. CONCLUSIONES

📖 Se concluyó que el contenido de un curso de control difuso debe cubrir como mínimo

los siguientes aspectos:

- ✓ Introducción al control convencional.
- ✓ Definición de los conceptos de lógica difusa.
- ✓ Aplicación de la lógica difusa al control.
- ✓ Herramientas para el diseño de controladores difusos

📖 Debido a que el control difuso aún no cuenta con una teoría para su análisis desde el punto de vista matemático, es necesario que el curso desarrolle prácticas de simulación e implementación para la verificación de los sistemas de control difuso diseñados.

📖 El gran desarrollo que ha logrado el control difuso se puede atribuir en un alto porcentaje a que no excluye las técnicas de control convencional si no que por el contrario, las refuerza en lo que se conoce como controladores híbridos (PID difuso, control adaptativo difuso, control difuso supervisorio, control difuso por modos deslizantes, etc.).

📖 Luego del análisis bibliográfico realizado durante esta investigación se determina la necesidad de continuar desarrollando trabajos en el área de control difuso, que permitan

mantener una constante actualización y enriquecer la escasa bibliografía existente en nuestro idioma al respecto.

📖 El presente trabajo puede ser usado como texto guía para el desarrollo de un curso de control difuso a nivel de pregrado o posgrado haciendo en este último un mayor énfasis en la implementación de sistemas de control difuso en procesos reales.

📖 Dada la similitud que tiene el control difuso con la manera intuitiva en que el ser humano realiza control y la sencillez y claridad con que se manejan los conceptos teóricos en este trabajo, éste puede ser abordado por cualquier estudiante que tenga bases mínimas sobre control de procesos.

📖 Como conclusión final resaltamos que el objetivo propuesto de diseñar el contenido de un curso de control difuso se alcanzó y se materializa en la presente monografía, además se incluyen 4 anexos como ayudas pedagógicas así: prácticas de simulación, prácticas de laboratorio, ejercicios y problemas propuestos, una presentación en diapositivas para uso dentro de las clases magistrales y un CD con el material antes citado y los demos del software del capítulo 7.

📖 Se concluye dado el gran aporte que tuvo para el desarrollo de este trabajo los servicios de la Internet (E-mail, FTP, paginas Web.), que cualquier trabajo de investigación moderno debe tener como herramienta de consulta obligatoria la Internet.

BIBLIOGRAFIA

BABUSKA R. y VERBRUGGEN H.B. An overview of fuzzy modeling for control. En: Control Engineering Practice. Vol. 4 No. 11; p. 1593-1606.

CHEN Chie-Li y CHANG Ming-Hui. Optimal Design of Fuzzy Sliding-mode Control: A Comparative study. En: Fuzzy Sets and Systems. 93 (1998); p. 37-48. Elsevier.

C.H. Chen. Fuzzy Logic and Neural Network HandBook. Editorial McGraw-Hill, Inc. New York 1996.

CHO,Hyun. CHO, Kwang y Wang, Bo. Fuzzy-PID hybrid control: Automatic rule generation using genetic algorithms. En: Fuzzy Sets and Systems, Vol. 92; p. 305-316

CHUNG, CHEN y LIN. A PI-type fuzzy controller with self-tuning scaling factors. En: Fuzzy Sets and Systems, Vol. 93; p. 23-28. Elsevier.

DELGADO, VERDEGAY, VILA. Breve Historia de la Lógica Fuzzy. Arbor septiembre-octubre (pp. 19-34.) 1993 En: Minicurso de Control Inteligente compilado por Freddy Naranjo. ACA, CUAO. Cali 1994.

FARINWATA, Shehu. On the Stability of Fuzzy Control Rulebase for a Nonlinear Process. En: Proceedings of Third IEEE International Conference On Fuzzy Systems. 1994; p. 924-929.

FILER y YAGER., Sliding mode design of fuzzy logic controllers. En: Dynamic Systems and control: Vol 1 ASME 1994; p 441-446.

FU, Li Min. Neural Networks in Computer Intelligence. McGraw Hill. 1994. 460 p.

GUELLEY Ned, JANG Roger, Fuzzy logic Toolbox for Use with Matlab. The Math Works Inc. 1995.

HABER, Rodolfo., Curso Introduccion al Control Difuso (Fuzzy) CUAO, Centro de Investigaciones, Marzo de 1995 Cali.

HARRISON, H. L., BOLLINGER John G., Controles Automáticos, editorial Trillas México 1983.

HAYKIN, Simons. Neural Networks A comprehensive foundation. Macmillan. 1994. 696 p.

HOGANS, HOMAIFAR y SAYYARRODSARI. Fuzzy Inference for Variable Structure Control En: Journal of Intelligent and Fuzzy Systems. Vol. 2. John Willey e Hijos. 1994; p. 229-241

HUANG, Liang J., TOMIZUKA Masayoshia. A Self-Paced Fuzzy Tracking Controller for Two-Dimenssional Motion Control En: IEEE Transaction on Systems Man. And Cybernetics, Vol. 20, No. 5; p. 1115-1123 1990

J.Y Hung, W. Gao y J.C. Hung., Variable structure control: a survey. En: IEEE Transactions on industrial electronics. Vol 40 No. 1; p. 2-14.

JAMSHIDE, VADIEE, ROSS. Fuzzy Logic and Control Software and Hardware applications. Editorial PTR Prentice Hall. New York 1993.

KAWAJI S. y MATSUNAGA N. Fuzzy Control of VSS Type and its Robustness. En: Fuzzy Control Systems. Editado por Kandel A. y Langholz G. CRC Press; p. 225-242.

LANGARI R. y YEN J. Introduction To Fuzzy Logic Control. En: Industrial Applications of Fuzzy Logic and Intelligent Systems. Editado por: Yen, Langari y Zadeh. IEEE Press; p. 3-39.

LANGARI y TOMIZUKA. Stability of Fuzzy Linguistic Control Systems. En: Proceedings of 29TH IEEE Conference on Decision and Control. 1990; p. 2185-2190.

LEE, Chuen C. Fuzzy logic in control systems:fuzzy logic controller. Parte I y II. En: IEEE transactions on systems,man and cybernetics. Vol. 20. No. 2

LIN Cheng y CHEN Yaw., Design of adaptive fuzzy sliding mode for nonlinear system control En: Fuzz IEEE 1994; p. 2-14.

MAMDANI, E. H. Twenty years of fuzzy control: experiences gained and lessons learnt. En : Fuzzy Logic : Technology and applications. Editor Robert Marks II. IEEE; p. 19-24.

MAN, TANG y KWONG. Genetic Algorithms: Concepts and Applications. En: IEEE Transactions on Industrial Electronics. Vol. 43 No. 5. 1996; p. 519-531.

MARTIN F. y THRO E. Fuzzy Logic A practical Approach. AP Professional. 292 p.

NARANJO PEREZ, Freddy. Estabilidad de Sistemas Lineales con Parámetros constantes y con Parámetros Inciertos. Centro de Investigaciones Universidad Autónoma de Occidente. Cali. Nov. 1994.

NARANJO PEREZ, Freddy. Control de estructura variable en tiempo discreto : comparación de algoritmos. En: 7° Congreso Latinoamericano de control automático. LACC-IFAC. Buenos Aires 1996; p. 1040-1045.

OGATA, Katsuhiko. Ingenieria de Control Moderna. Segunda Edición. Prentice Hall. 1993.

PALM, Rainer. Robust control by fuzzy sliding mode. En: Automática. Vol. 30 No. 9; p. 1429-1437.

PASSINO, Kevin. Bridging the gap perceived between conventional and intelligent control. En: IEEE control systems. Junio 1993; p. 12-18.

PASSINO, Kevin, ORDOÑEZ, Raul ZUMBERGE, Jon y SPOONER, Jeffrey T Adaptive Fuzzy Control : Experiments and Comparative Analyses,. En: IEEE Transactions on Fuzzy Systems, Vol. 5, No 2, Mayo de 1997.

PEDRYCS, Witold. Fuzzy control engineering: reality and Challenges. En: Fuzz IEEE 1995; p. 437-446.

PEDRYCZ, Wiltold. Fuzzy Sets Engineering Editorial CRC Press, Inc. 1995

PEREIRA John y BOWLES Jhon. A comparison of PID y Fuzzy Control of a Model Car. En: IEEE Journal of Intelligents and Fuzzy Systems, Octubre de 1994; p. 849-854

PERRY, Tekla S., Then inventor of fuzzy logic persisted despite decades of opposition
LOTFI A. ZADEH En: IEEE Spectrum. Junio 1995; p. 32-35.

RAYCHAUDHRI y BELL. From Conventional Control to Autonomous Intelligens Methods. En: IEEE Control Systems. Octubre 1996; p. 78-83.

ROSS, Tymothy J., Fuzzy Logic With Engineering Applications Editorial McGraw-Hill, Inc. New York 1997.

SHOURESHI, Rahmat. Intelligent Control Systems: Are They for Real.? En: Journal of dynamic Systems, Measurement and Control. Junio 1993. Vol. 115; p. 392-400.

VERBRUGGEN H, KRIJGSMAN A. y BRUISN P. Towards Intelligent Control: Integration of AI in Control. En: Applications of Artificial Intelligence in Process control. Editado por Boullart L., Krijgsman A. y Vingerhoeds R. Pergamon Press; p. 223-249.

VERDEGAY, José L. Introduccion al Control Difuso En: Memorias Primer Congreso de la ACA, Noviembre de 1994.

VON ALTROCK, Constantin. Fuzzy Logic and Neurofuzzy Applications Explained. Printice Hall. 1995. 3548 p.

WANG, Li-Xin Adaptative Fuzzy Systems and Stability Analysis. Editorial PTR Prentice Hall. New York 1994.

WANG, Li-Xin A Supervisory Controller for Fuzzy Control Systems That Gurantees Stability. En: Proceedings of Third IEEE International Conference On Fuzzy Systems. 1994; p. 1035-1039.

WITOLD, Pedrycz. Fuzzy Sets Engineering. Editorial CRC Press, Inc Florida 1995

YEN, Jhon. Industrial applicationss of fuzzy logic and intelligent systems IEEE 1995.

YI Soon y CHUNG Myung. Stability Analysis of a Fuzzy Logic Controller for an Uncertain Dynamic System. En: Proceedings of Third IEEE International Conference On Fuzzy Systems. 1994; p. 1028-1039.

ZADEH, Lotfi. Fuzzy Sets En: Information and Control, Vol. 8; 1965 p. 338-353.

ZADEH, Lotfi. The Evolution of Systems Analysis and Control: A personal Perspective En: IEEE Control Systems. Junio 1996; p. 95-98.

ANEXO A. Prácticas de simulación

1. Prácticas de simulación de control difuso.

1.1. Práctica No 1 : Manejo del ToolBox de Fuzzy de Matlab.

1.1.1. Objetivo General

Conocer el manejo del software de simulación ToolBox Fuzzy de Matlab.

1.1.2. Objetivos Específicos

- Instruir al estudiante en el manejo de los GUI de Toolbox de Fuzzy.
- Enseñar la metodología a seguir para implementar un sistema usando Simulink y el Toolbox de Fuzzy y Simulink

1.1.3. Manejo del Toolbox de Fuzzy

Como se vio en el capítulo 7, el ToolBox es una colección de funciones construidas en el entorno de cálculo numérico de Matlab. Su manejo se puede hacer de dos formas usando directamente sus funciones desde MatLab en la línea de comandos o usando los GUI (interface gráfica de usuario). Por cualquiera de los dos métodos podemos implementar un sistema difuso (funciones de pertenencia, reglas, fuzificación, etc.) que posteriormente para efectos de simulación puede ser llevado a Simulink para validar su funcionamiento, o implementarse usando el Hardware adecuado. El uso de Toolbox con Simulink permite trabajar sistemas difusos Mamdani o Sugeno creados con FIS en el entorno de simulación de Simulink, el cual trabaja a través de bloques a los cuales se les dan sus respectivos parámetros para el proceso de simulación, tiene características tales como permitir la animación, bloques generadores de señales y la visualización gráfica de entradas y salidas.

Debido a que en capítulo 7 se trató con bastante profundidad el ToolBox, nos limitaremos a indicar los pasos a seguir para implementar un sistema difuso y luego simularlo con la ayuda de Simulink. Se usarán las interfaces gráficas GUI ya que son mucho más intuitivas que trabajar en las líneas de comandos de MatLab.

En la figura A-1 podemos ver los pasos para implementar un controlador difuso con ayuda Matlab, que de manera general fueron mencionados en el capítulo 7.

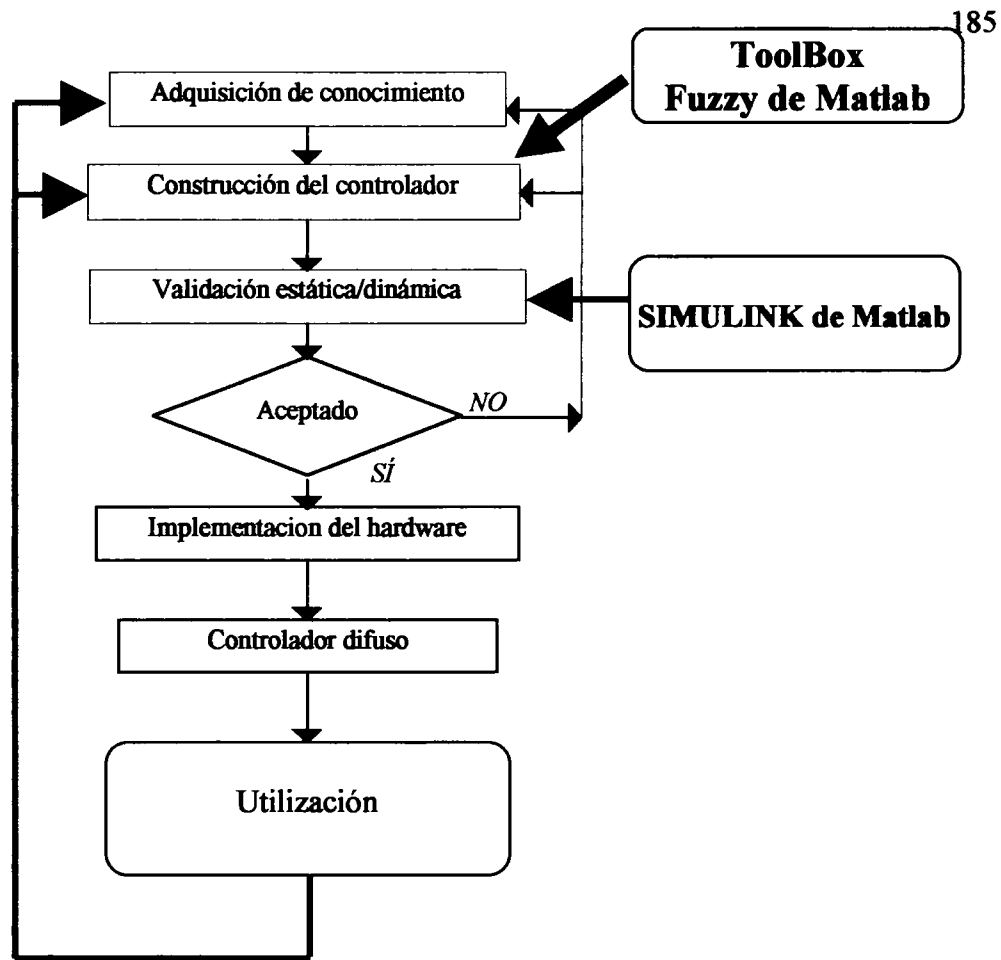


Figura A-1 Desarrollo de un controlador difuso (FLC) usando Matlab

Tomando como referencia la figura A-1 se definen los siguientes pasos para el desarrollo de la estrategia control lingüística basada en reglas (control difuso) a simular:

1. Descripción del problema de control
2. Descripción lingüística de la estrategia de control
3. Definición de variables lingüísticas de entrada y salida
4. Fuzzificación de las variables de entrada
5. Definición del motor de inferencia
6. Defuzzificación de las variables lingüísticas de salida
7. Simulación y validación

El primer paso nos permite definir el problema de control y nos da los elementos para que a través de la heurística se logre desarrollar la estrategia de control.

Para el desarrollo de la práctica el estudiante debe preparar un modelo difuso que controle una planta y haber desarrollado los seis pasos antes mencionados.

Los elementos conceptuales para la definición de las variables lingüísticas, la fuzzificación, el método de inferencia y la defuzzificación fueron tratados en el capítulo 4.

Una vez desarrollados los primeros seis pasos para el proceso a controlar podemos entrar a crear nuestro sistema en el Toolbox, para ello debemos cargar el editor FIS digitando *fuzzy* en la línea de comandos de Matlab. La cual cargará la siguiente imagen :

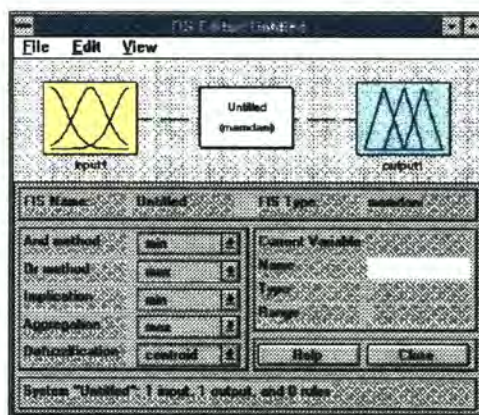


Figura A-2 Editor FIS

Por defecto el sistema a implementar en el Toolbox es de tipo Mamdani pero si desea uno tipo Sugeno lo puede hacer desde la opción **File**. Una vez definido el tipo de sistema y

usando la opción **Edit** del menú, cree las variables de entrada de su sistema (puede crear sistemas MIMO). Recuerde que en esta opción se encuentran las opciones para editar tanto las variables de entrada como de salida. Para esta práctica utilizaremos un sistema tipo Mamdani.

Asigne los nombres adecuados a cada variable dando click en cada una y digitando el nombre en la casilla de **Name**.

De doble click en cada una de las variables para asignarles sus funciones de pertenencia usando la opción **Edit** del menú (el editor inmediatamente cambia a editor de funciones de pertenencia). En esta opción se tienen 8 diferentes tipos de funciones de pertenencia a escoger pero se entiende que si siguió los seis pasos citados anteriormente ya tiene claro qué tipo de funciones usará. Ver figura A-3

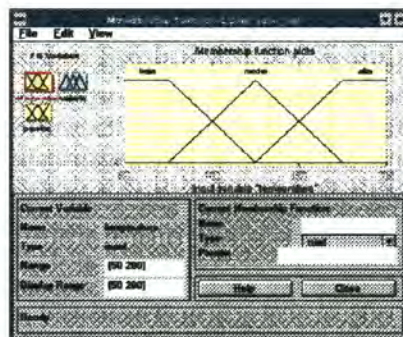


Figura A-3 Editor de funciones de pertenencia.

Establezca los dominios para todas las funciones de pertenencia. Defina luego los métodos AND, OR y implicación, agregación y defuzzificación. Usando la opción **View** active el

editor de reglas para que pueda establecer la base de reglas para su sistema. En el editor de reglas existe la opción para elaborar reglas: verbalmente, simbólicamente o indexado. La primera es la más intuitiva.



Figura A-4 Editor de reglas.

Veamos la base de reglas para nuestro un ejemplo usando los tres formatos:

Formato simbólico (symbolic)

1. (temperatura==baja) & (presion==baja) => (valvula=abrir) (1)
2. (temperatura==baja) & (presion==media) => (valvula=abrir) (1)
3. (temperatura==baja) & (presion==alta) => (valvula=no_operar) (0.2)
4. (temperatura==media) & (presion==baja) => (valvula=abrir) (1)
5. (temperatura==media) & (presion==media) => (valvula=no_operar) (1)
6. (temperatura==media) & (presion==alta) => (valvula=cerrar) (0.2)
7. (temperatura==alta) & (presion==baja) => (valvula=no_operar) (1)
8. (temperatura==alta) & (presion==media) => (valvula=cerrar) (1)
9. (temperatura==alta) & (presion==alta) => (valvula=cerrar) (0.2)

Formato Verbal ingles(verbose)

1. If (temperatura is baja) and (presión is baja) ten (válvula is abrir) (1)
2. If (temperatura is baja) and (presion is media) ten (válvula is abrir) (1)
3. If (temperatura is baja) and (presion is alta) ten (válvula is no_operar) (0.2)
4. If (temperatura is media) and (presion is baja) ten (válvula is abrir) (1)
5. If (temperatura is media) and (presion is media) ten (válvula is no_operar) (1)
6. If (temperatura is media) and (presion is alta) ten (válvula is cerrar) (0.2)
7. If (temperatura is alta) and (presion is baja) ten (válvula is no_operar) (1)
8. If (temperatura is alta) and (presion is media) ten (válvula is cerrar) (1)
9. If (temperatura is alta) and (presion is alta) ten (válvula is cerrar) (0.2)

Formato Verbal frances(verbose)

1. Si (temperatura est baja) et (presion est baja) alors (válvula est abrir) (1)
2. Si (temperatura est baja) et (presion est media) alors (válvula est abrir) (1)
3. Si (temperatura est baja) et (presion est alta) alors (válvula est no_operar) (0.2)
4. Si (temperatura est media) et (presion est baja) alors (válvula est abrir) (1)
5. Si (temperatura est media) et (presion est media) alors (válvula est no_operar) (1)
6. Si (temperatura est media) et (presion est alta) alors (válvula est cerrar) (0.2)
7. Si (temperatura est alta) et (presion est baja) alors (válvula est no_operar) (1)
8. Si (temperatura est alta) et (presion est media) alors (válvula est cerrar) (1)
9. Si (temperatura est alta) et (presion est alta) alors (válvula est cerrar) (0.2)

Formato indexado.

1 2, 2 (1) : 1
 1 1, 2 (1) : 1
 1 3, 3 (0.2) : 1
 2 2, 2 (1) : 1
 2 1, 3 (1) : 1
 2 3, 1 (0.2) : 1
 3 2, 3 (1) : 1
 3 1, 1 (1) : 1
 3 3, 1 (0.2) : 1

Para efectos de sintonía puede modificar los pesos que corresponden al último número de la derecha.



Figura A-5 Visualizador de activación de reglas

Una vez definida la base de reglas puede desde la opción **View** activar el visualizador de activación de reglas (ver figura A-5), que le permite con el Mouse variar los valores de las entradas y ver el comportamiento de cada regla y el resultado de la defuzzificación. De igual manera puede visualizar la superficie generada por el sistema como se ve en la figura A-6. Con estas opciones puede empezar a simular la respuesta del sistema ante entradas puntuales.



Figura A-6 Visualizador de la superficie de decisión o de inferencia

Si desea realizar la simulación completa debe entonces grabar su sistema desde la opción **File** y **Save** o **Save As**, después cargue el Simulink digitando **Simulink** desde la línea de comandos de Matlab. Luego inserte el bloque de control el cual tendrá sus respectivas entradas y salidas de acuerdo con la definición en el FIS. Para traer el bloque de control digite **fuzblock** en la línea de comandos que mostrará el bloque que se ve a continuación



Figura A-7 Controlador difuso para Matlab.

También puede copiar el bloque de los archivos de ejemplo que trae el Toolbox `sltank`.

Para que el bloque de control cargue su sistema debe dar click en él y digitar el nombre del archivo `.FIS` que grabó anteriormente, pero antes debe cargarlo en el espacio de trabajo del Matlab. El siguiente paso es con ayuda de las herramientas de simulink diseñar los bloques que representen el modelo identificado de la planta a controlar. Después se conectan los bloques de generación de señales (escalón, rampa, etc.) y de graficación al controlador y al modelo de la planta, para poder simular el desempeño del controlador difuso como podemos ver en la siguiente gráfica:

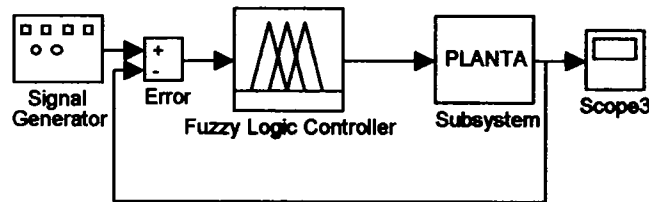


Figura A-8 Esquema de control difuso con Simulink.

Para efecto de análisis se pueden variar el tiempo de simulación al igual que la tolerancia y el método numérico que usa Simulink.

Finalmente y después del análisis de resultados se procede a sintonizar el controlador difuso modificando este con ayuda de los GUI o la línea de comandos de Matlab y se repite la simulación hasta obtener la sintonía deseada.

Ejercicios sugeridos

- Implemente un sistema SISO y uno MIMO con ayuda del ToolBox y simúlelos con el Simulink
- Desarrolle un sistema difuso que contenga todas las funciones de pertenencia habilitadas del Toolbox asegurándose de conocer los parámetros de cada una.
- Elabore una base de reglas de prueba en los diferentes formatos, varíe luego las reglas y analice la activación de las mismas con el visualizador de activación de reglas.

1.2. Práctica No 2 : Simulación del Control de un Servomotor

1.2.1. Objetivo General

Generar destreza en el estudiante para el manejo del ToolBox de Fuzzy.

1.2.2 Objetivos específicos

- Simular el modelo de un servomotor
- Controlar un servomotor usando el Toolbox Fuzzy de Matlab.

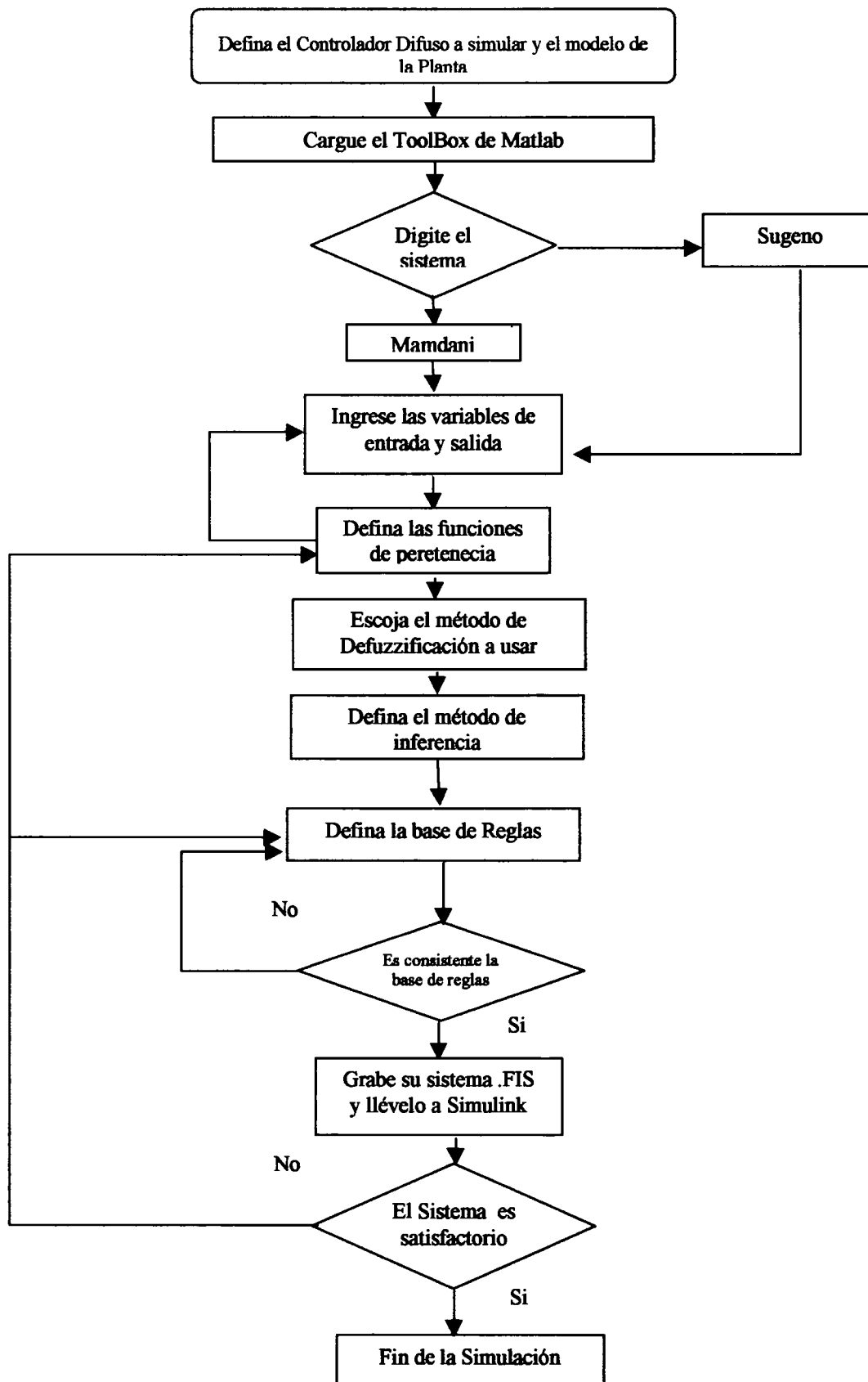


Figura A-9 Diagrama de flujo para simulación.

1.2.3 Desarrollo

Los servomotores son de alto uso en la industria y presentan una serie de no linealidades como zonas muertas y fricción de la carga entre otras, por este motivo se plantea realizar el control de un servomotor siguiendo la metodología descrita en la práctica No 1. y resumida en la figura A-9

Elabore un sistema difuso con ayuda del toolbox para controlar un servomotor usando el error y la derivada del error como variables de entrada y el voltaje de armadura como variable de salida. Genere el archivo.FIS del controlador difuso y llévelo al espacio de trabajo de Matlab, para luego insertarlo como un bloque de control difuso en Simulink como muestran las figuras A-10 y A-11.

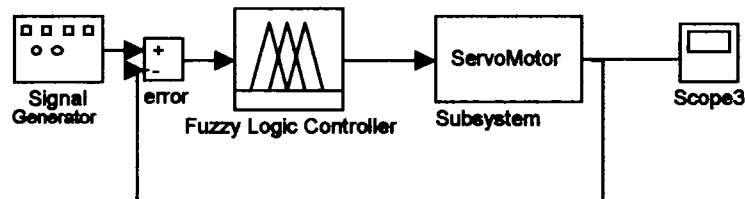


Figura A-10 Controlador difuso para un servomotor.

Implemente el modelo del motor usando la siguiente ecuación A-1 en Simulink, introduzca una no linealidad usando una zona muerta de 0.25 segundos.

$$E_a(S) = RI_a(S) + LSI_a(S) + K_e S\theta(S)$$

$$T = K_t I_a(S)$$

$$T = I_m S^2 \theta(S) + b S \theta(S)$$

$$\frac{\theta(S)}{E_a(S)} = \frac{K}{S(\tau S + 1)}$$

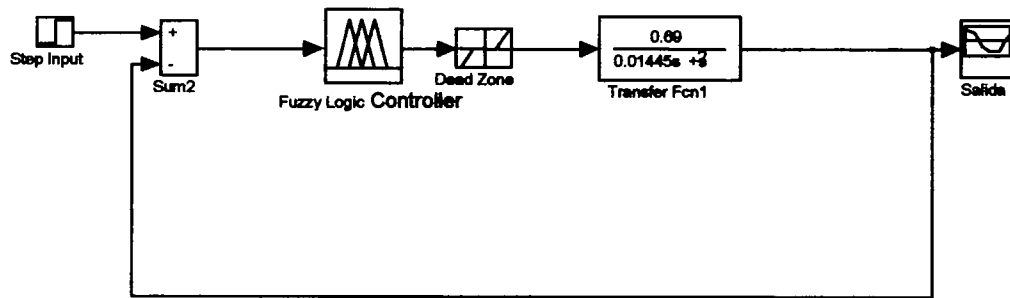


Figura A-11 Simulación para el modelo del servomotor

Para implementar el modelo del motor utilice la siguiente ecuación (A-2):

$$\frac{\theta(S)}{E_a(S)} = \frac{0.69}{S(0.01445S + 1)} \quad (\text{A-2})$$

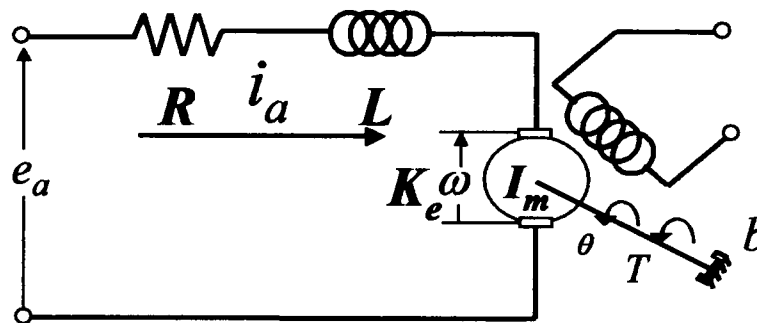


Figura A-12 Modelo del servomotor

- ☐ Verifique la respuesta a un escalón.
- ☐ Sintonice el controlador variando los soportes de los conjuntos difusos, funciones de pertenencia, el método de defuzzificación y la base de reglas y repita la simulación.

- Repita la misma simulación para un modelo con los siguientes parámetros:

$$L = 0.0004 \text{ H}$$

$$R = 1.6 \Omega$$

$$b = 0.0007639 \text{ Nms/rad}^2$$

$$K_e = 0.488 \text{ Nm/A}$$

$$K_t = 0.488 \text{ Nm/A}$$

- Agregue una perturbación al sistema usando una de las fuentes de señales de Simulink, haga su análisis del comportamiento del CD.
- Diseñe un PID que ante una entrada de un escalón haga que el error en estado estacionario sea cero y el mínimo sobrepico posible, compare este con un CD que tenga las siguientes base de reglas:

| <i>de/e</i> | NL | NM | NS | ZE | PS | PM | PL |
|-------------|----|----|----|----|----|----|----|
| NL | NX | NX | NX | NX | NS | ZE | PX |
| NM | NX | NX | NL | NL | NS | PS | PX |
| NS | NX | NL | NL | NM | PS | PS | PX |
| ZE | NX | NL | NL | ZE | PL | PL | PX |
| PS | NX | NS | NS | PM | PL | PL | PX |
| PM | NX | NS | PS | PL | PL | PX | PX |
| PL | NX | ZE | PS | PX | PX | PX | PX |

Donde e es el error y de la derivada del error y los conjuntos difusos: NL (negativo largo), NM (negativo medio), NS (negativo pequeño), ZE (Zero), PS (positivo pequeño), PM (positivo medio), PL (positivo largo), iguales para cada variable, cambiando sólo su dominio. Use -1 y 1 para el error y -6 y 6 para la derivada del error. El método de

defuzzificación queda a su elección, al igual que los tipos de funciones de pertenencia de e y de . La variable de salida es el voltaje del amplificador cuyo dominio es -50 y 50 con conjuntos difusos: NL (negativo largo), NX (negativo intermedio), NS (negativo pequeño), ZE (Cero), PS (positivo pequeño), PX (positivo intermedio), PL (positivo largo), las funciones de pertenencia deben ser singleton. Después compare los resultados de los dos controladores.

☐ Realice su análisis y dé sus conclusiones.

Anexo B: Práctica de Laboratorio

1. Práctica de Laboratorio de control difuso.

1.1. Práctica No 1: Control de Velocidad en un motor DC.

1.1.1. Objetivo General

Implementar en el laboratorio un controlador difuso para el control de un motor DC

1.1.2. Objetivos Específicos

- Instruir al estudiante en el manejo del hardware para la implementación de controladores Difusos.
- Brindar al estudiante las herramientas para que sea capaz de controlar a través de un computador un motor DC.
- Comprobar las simulaciones efectuadas en el anexo A.

1.1.3. Práctica de Laboratorio: Control difuso de un motor DC.

Equipo requerido :

- Computador Personal
- Motor DC
- Amplificador de voltaje
- Tarjeta Análogo digital
- Software de control Difuso (SCD)
- Tacogenerador
- Conectores

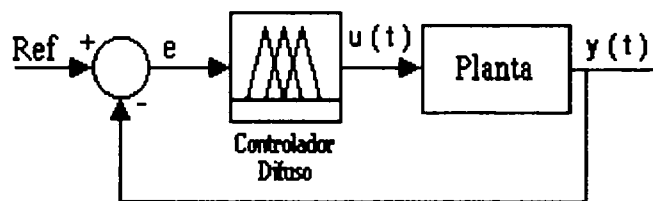


Figura B-1 Esquema de CDD

Descripción : La práctica persigue que a través del software SCD¹⁹, se controle la velocidad de un motor DC. Este software permite implementar en un computador personal (PC) controladores difusos directos (CDD) dentro del esquema clásico de control donde el controlador recibe el error y en función de este determina la acción de control ver figura B-1.

¹⁹ Software desarrollado como proyecto de grado por Oscar Mauricio Agudelo, CUAO. 1997

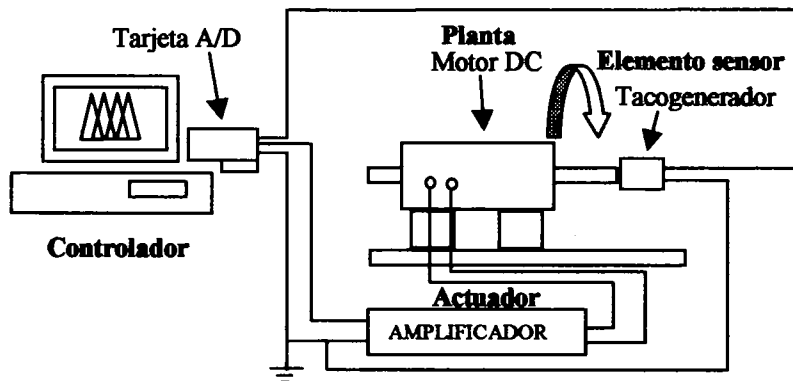


Figura B-2 Diagrama del montaje

El tipo de controlador a implementar es de tipo Mamdani, con funciones de pertenencia triangulares y trapezoidales, la evaluación de reglas se efectúa usando el MAX-MIN. Como método de defuzzificación se usa el centroide o centro de masa.

La planta la constituye el motor DC, el actuador está dado por el amplificador y el elemento sensor es el tacogenerador, ver figura B-2.

Características del montaje:

- El tacogenerador entrega 7V cuando el motor se encuentra girando a 1000 RPM (Revoluciones por minuto).
- El motor es controlado por el voltaje de armadura.

Pasos para la creación del sistema: Con base en modelo matemático mostrada en la figura B-3 (obtenido mediante identificación del motor del laboratorio) y usando el esquema de simulación del anexo A, proceda a implementar su controlador difuso usando el SCD como se indica a continuación:

- Cargue el programa SCD.
- Cree un nuevo sistema (opción Fichero – Nuevo)
- Defina las entradas del controlador Difuso (error, derivada del error, integral del error o segunda derivada del error)
- Defina la salida del controlador Difuso
- Defina las funciones de pertenencia de las entradas y la salida. Recuerde que el programa puede trabajar solamente con funciones de pertenencia triangulares y trapezoidales.
- Defina la base de reglas.
- Pruebe el controlador difuso con la opción que tiene el software para este fin.

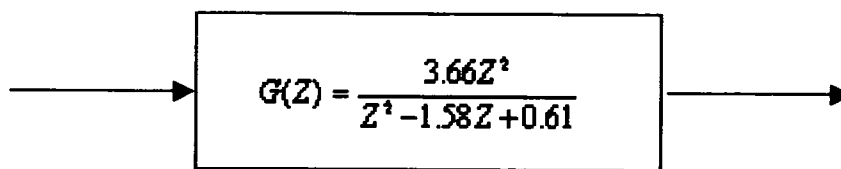


Figura B-3 modelo matemático discreto del motor a controlar.

Configuración de la tarjeta: Ajuste los siguientes parámetros de la tarjeta de adquisición de datos.

- **Tarjeta por puerto paralelo TDAD-800 (8 bits):** Se debe configurar la dirección base del puerto paralelo. (Use la dirección 378H). Configure los canales de entrada y salida que pueden ser bipolares o no. Para esta tarjeta se deben usar los siguientes:

ENTRADA

Unipolar : 0V -> 5V

Bipolar : -2.5V -> 2.5V

SALIDA

Unipolar: 0V -> 5V

Bipolar: -5V -> 5V

- **Tarjeta DAS - 1600 (12 bits):** La dirección base también es 378H. Los canales de entrada deben estar configurados en modo "no diferencial". Escoja cualquiera de los 16 canales ("no diferenciales") disponibles en la tarjeta (0 - 15) como entrada. Para la salida uno de los dos canales disponibles (0 - 1).

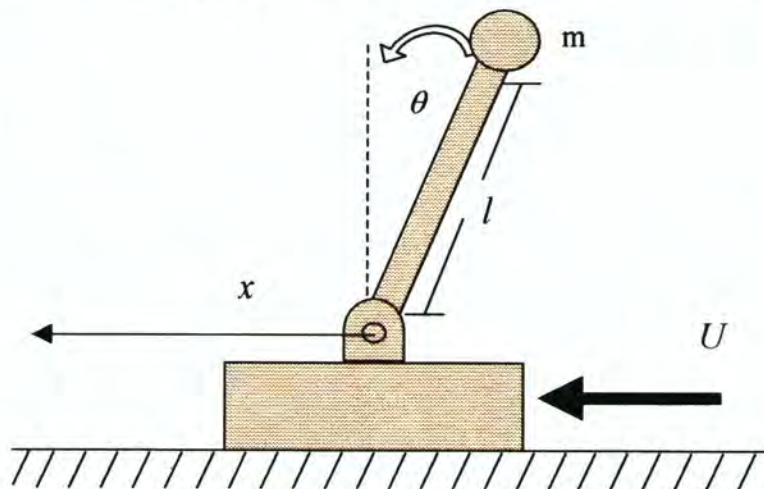
Desarrollo : Una vez cargado el controlador difuso en el SCD y ajustados los parámetros de la tarjeta, proceda a realizar las conexiones como muestra la figura B-1. Verifique la polaridad de las conexiones ya que esto podría poner en riesgo su seguridad y la de los equipos. Proceda a ejecutar el programa. Efectúe las siguientes operaciones :

- Verifique el sistema con un filtro para la señal de Tacogenerador
- Varíe el Set-point.
- Modifique las reglas.
- Analice la respuesta del controlador y de sus conclusiones.

1.14. Práctica opcionales: Debido a que el laboratorio de control de la universidad cuenta con otras plantas como los sistemas placa ventilador, control de presión, etc. Se deja abierta la posibilidad para que bajo el mismo esquema de la Práctica de control del motor DC el estudiante implemente mas Prácticas con el fin de afianzar su conocimiento en control difuso. Para estas Prácticas opcionales se recomienda simular siempre el sistema en el Toolbox de Matlab (anexo A) y luego si proceder a la implementación en el laboratorio.

Anexo C: Ejercicios y problemas.**Taller Capítulo I**

- 1.1 Explique en qué consiste “el problema de control”.
- 1.2 Defina las características del control clásico.
- 1.3 Halle el modelo matemático para el sistema del péndulo invertido de la figura C-1.

**Figura C-1** Péndulo invertido

- 1.4 Identifique las herramientas de análisis mas importantes en control clásico.

1.5 Al diseñar controladores se desea que estos cumplan ciertas especificaciones dentro de dos dominios posibles. Elabore una tabla donde se identifique estos dos dominios y las especificaciones de cada uno.

1.6 Cuáles son los compensadores mas comúnmente utilizados.

1.7 Diga cuáles son las características del control moderno.

1.8 Encuentre las variables de estado y la función de transferencia para el sistema del péndulo invertido de la figura C-1.

1.9 Indique Cuáles son las técnicas de control moderno mas difundidas.

1.10 Defina las características del control inteligente.

1.11 Cuáles son las técnicas de control inteligente más reconocidas.

1.12 Explique la aplicación en control de:

- Sistemas expertos.
- Redes Neuronales.
- Algoritmos Genéticos.

Taller Capítulo II

2.1 Qué es la lógica difusa.

2.2 Destaque los antecedentes más importantes en el desarrollo de la lógica difusa.

2.3 Explique en qué consiste el razonamiento aproximado y su relación con la lógica difusa.

2.4 Defina qué es un conjunto difuso.

2.5 Cuáles son las características de un conjunto difuso.

2.6 Qué es un singleton difuso.

2.7. Qué es un conjunto difuso normal.

2.8 Para los conjuntos difusos A y B de la figura C-2 halle:

- La unión
- La intersección
- El complemento de ambos conjuntos.

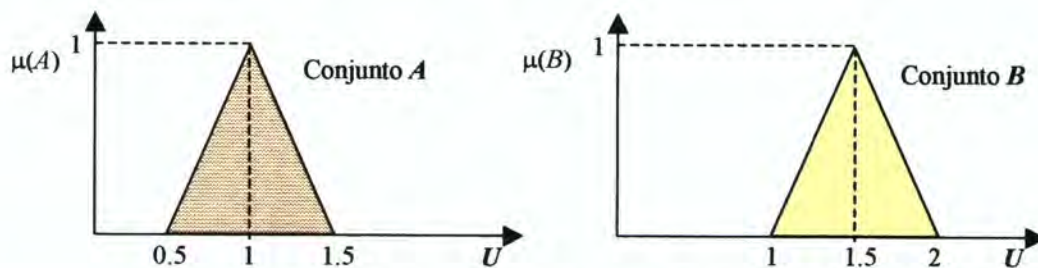


Figura C-2 Conjuntos difusos A y B .

2.9 Demuestre que el conjunto difuso Y está contenido en X ver figura C-3.

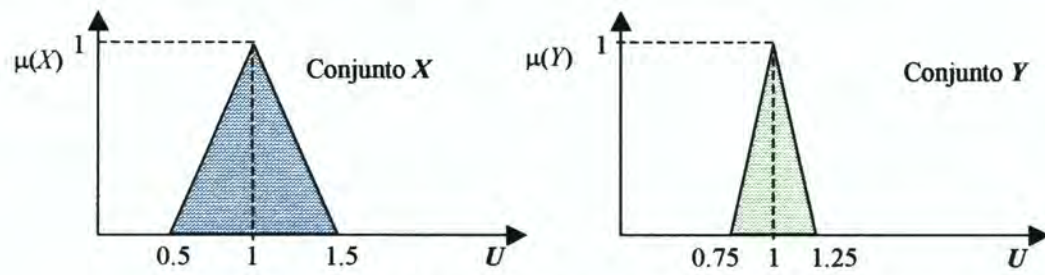


Figura C-3 Conjuntos difusos X y Y .

Taller Capítulo III

3.1 Dé la definición formal e intuitiva de variable lingüística.

3.2. Defina con todos sus elementos tres variables lingüísticas de posible aplicación en control (presión, velocidad, temperatura, nivel, etc.).

3.3 Qué es una regla difusa If-Then.

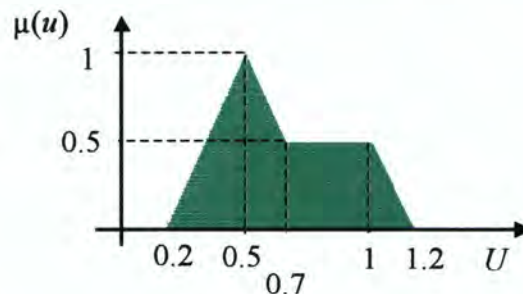
3.4 En qué consiste la fuzzificación.

3.5 Qué es una base de reglas y un motor de inferencias.

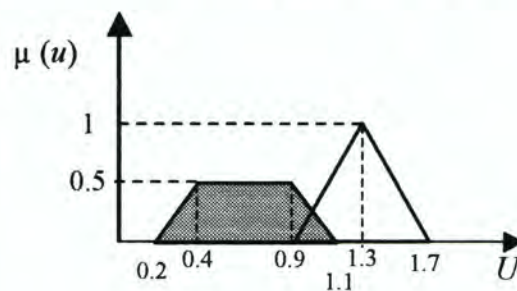
3.6 En qué consiste la defuzzificación.

3.7 Para cada uno de los siguientes conjuntos difusos realice su defuzzificación usando el método indicado.

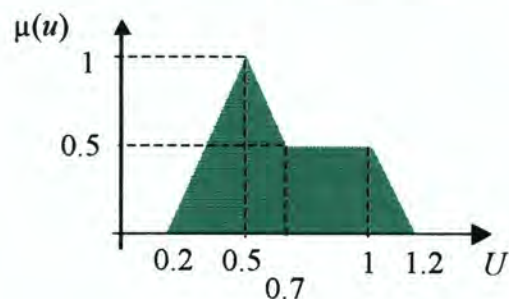
- Centro de gravedad



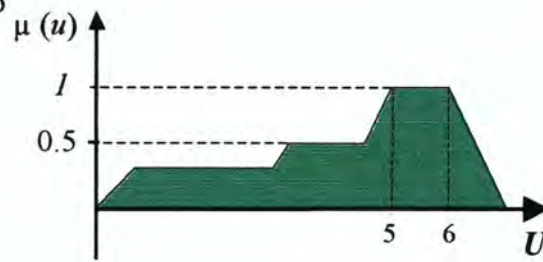
- Centro de las sumas



- Método de la altura



- Primer y último máximo



3.8 Qué es un sistema difuso

3.9 A qué se llama un sistema difuso Puro

3.10 Qué es un sistema tipo Takagi y Sugeno

3.11 Qué es un sistema tipo Mandani

3.12 Mencione algunas aplicaciones de lógica difusa.

Taller Capítulo IV

4.1 Defina el control difuso.

4.2 Cite cuatro de los desarrollos más importantes en el control difuso.

4.3 Dibuje la estructura básica de un controlador difuso.

4.4 Cuáles son las características de la interface de fuzzificación.

4.5 Para la variable lingüística velocidad mostrada en la figura C-4 defina la fuzzificación correspondiente de los siguientes valores de entrada: 10, 25, 55, 64, 78

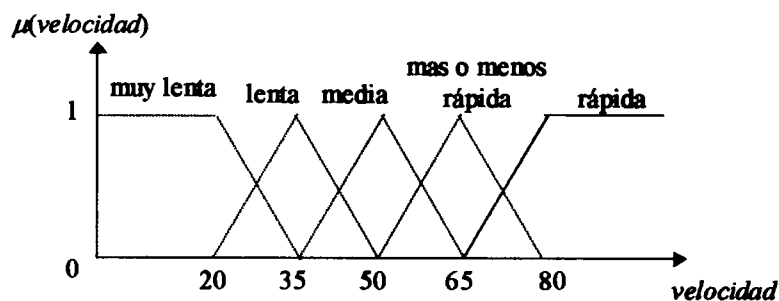


Figura C-4 Variable lingüística Velocidad.

4.6 En qué consiste la base de conocimiento de un Controlador difuso (CD).

4.7 En qué se diferencia la estrategia de agregación conjunción de la disyunción.

4.8 Qué papel desempeña el motor de inferencia en CD.

4.9 Defina la función de la interface de defuzzificación de un CD

4.10 Cuáles son las aplicaciones de los CD.

4.11 Dibuje el esquema de un CD directo.

4.12 Dibuje el esquema de un CD PID.

- 4.13 Explique la aplicación de CD y la técnica de modos deslizantes.
- 4.14 Dibuje el esquema de un CD adaptativo.
- 4.15 Explique las arquitecturas híbridas de CD más importantes.

Taller Capítulo V

5.1 Explique el concepto de estabilidad en CD.

5.2 Explique el concepto de robustez de CD.

Taller Capítulo VI

6.1 Investigue dos aplicaciones de lógica difusa en control (artículos de revistas, libros, etc.) e identifique cada una de sus partes. (prepare una exposición de estos).

Taller Capítulo VII

7.1 Defina Cuáles son las herramientas de software más importantes para el diseño y simulación de CD.

7.2 Desarrolle las prácticas indicadas en el Anexo A.

Taller Capítulo VIII

8.1 Defina Cuáles son los dispositivos de hardware más importantes para la implementación de CD.

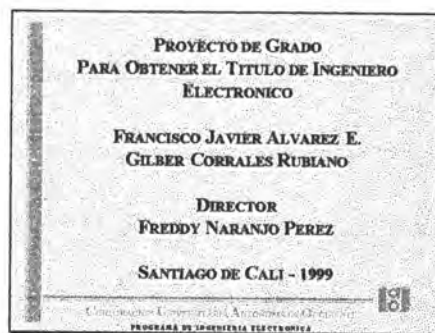
8.2 Desarrolle la práctica indicada en el Anexo B.

ANEXO D. Presentación del curso en diapositivas

Diapositiva 1



Diapositiva 2



Diapositiva 3

CAPITULO 1

El problema de control
Soluciones convencionales al problema de control:
 - Control Clásico
 - Control Moderno
Control Inteligente
Control Inteligente vs. Control Convencional
Técnicas de Control Inteligente:
 Sistemas Expertos
 Redes Neuronales
 Algoritmos Genéticos
 Lógica Difusa




Universidad de Occidente, Armenia, Colombia
 PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 4

El problema de control

Surge cuando al disponerse de un sistema (planta, proceso, máquina, etc.) queremos hacer que su comportamiento, que está reflejado en las variables que lo caracterizan físicamente, obedezca a nuestras intenciones, lo cual tiene necesariamente que coincidir con lo más conveniente a los fines con que ha sido creado.



Universidad de Occidente, Armenia, Colombia
 PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 5

Referencias Históricas

| | |
|--------------|---|
| <Siglo XVII | Maxwell realiza el primer modelo matemático |
| <30's y 40's | Transformada de Laplace, y H. Nyquist |
| <50's y 60's | Control óptimo y análisis de espacio de estados. Estabilidad de Liapunov. |
| <70's y 80's | Surgen avances en control estocástico, robusto y adaptativo |
| <90's | Control algebraico, por corriente y predictivo generalizado. |



Universidad de Occidente, Armenia, Colombia
 PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 6



Diapositiva 7



Diapositiva 8

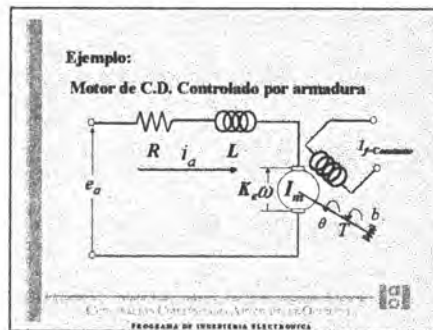


Diapositiva 9

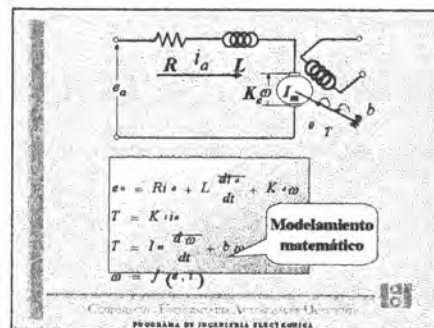
- Se utiliza extensamente el concepto de la función de transferencia.
- El análisis y diseño se realizan en el dominio de la variable de Laplace (S) y/o en el dominio de la frecuencia.
- Brinda muy buenos resultados en los sistemas de una entrada y una salida (SISO).

Comisión de Investigación Científica y Tecnológica
PROGRAMA DE INGENIERÍA ELECTRÓNICA

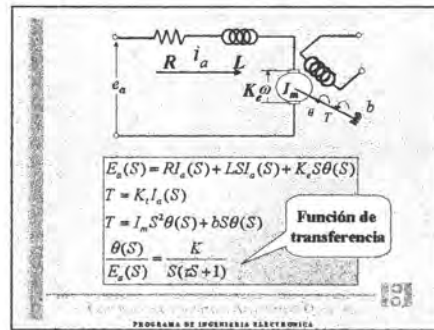
Diapositiva 10



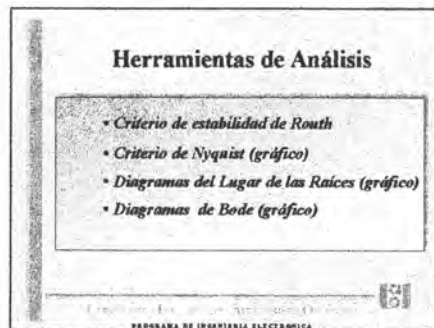
Diapositiva 11



Diapositiva 12



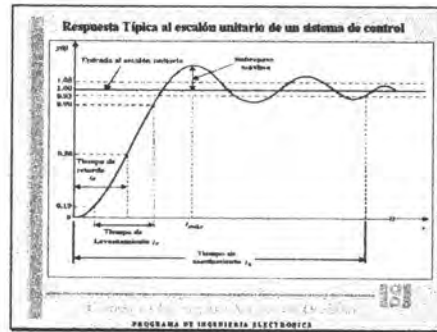
Diapositiva 13



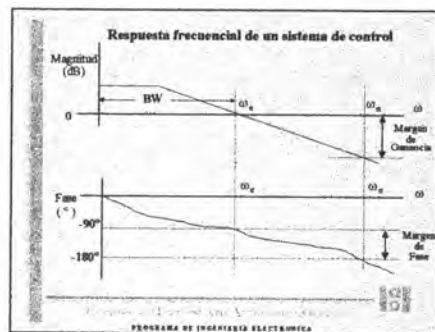
Diapositiva 14



Diapositiva 15



Diapositiva 16



Diapositiva 17

Los compensadores más comúnmente usados son:

- Compensador de factor de ganancia
- Compensador en adelanto
- Compensador en atraso
- Compensador en adelanto-atraso o atraso-adelanto
- PID y todas sus combinaciones

PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 18

Compensadores clásicos:

- $u = Ke$ (proporcional)
- $u = K_1 e + K_2 \int e dt$ (proporcional e integral)
- $u = K_1 e + K_2 \int e dt + K_3 \frac{de}{dt}$ (proporcional, integral y derivativo)
- $u = K_1 e + K_3 \frac{de}{dt}$ (proporcional y derivativo)
- $u = K \frac{s+a}{s+b}$ (adelanto o atraso de fase)

Universidad Politécnica de Valencia
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 19

Control moderno

Universidad Politécnica de Valencia
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 20

- Se basa en técnicas en el dominio del tiempo y permite resolver de la misma manera sistemas SISO y MIMO.
- Las técnicas de control moderno sirven tanto para análisis como para diseño.
- El modelamiento matemático de los sistemas se realiza en forma de ecuaciones de estado.
- La formulación matemática en el espacio de estados es fácilmente programable en un computador.

Universidad Politécnica de Valencia
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 21

Análisis y diseño mediante las técnicas de control moderno.

- *Estado*
- *Variable de estado*
- *Controlabilidad*
- *Observabilidad*

PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 22

Variables de estado

$$\frac{d^n x}{dt^n} + a_{n-1} \frac{d^{n-1} x}{dt^{n-1}} + \dots + a_1 \frac{dx}{dt} + a_0 x = b_0 u$$

$$x_1 = x, x_2 = \dot{x}, \dots, x_n = \frac{d^{n-1} x}{dt^{n-1}}$$

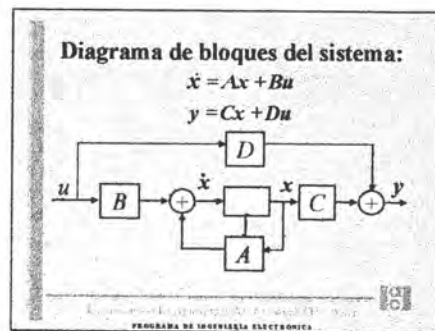
$$\begin{aligned} \dot{x}_1} &= x_2 \\ \dot{x}_2} &= x_3 \\ &\vdots \\ \dot{x}_{n-1} &= x_n \\ \dot{x}_n &= -a_0 x_1 - a_1 x_2 - \dots - a_{n-1} x_n + b_0 u \end{aligned}$$

Forma general

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 23



Diapositiva 24

Función de transferencia

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t)$$

$$sX(s) = AX(s) + BU(s)$$

$$Y(s) = CX(s)$$

$$\frac{Y(s)}{U(s)} = G(s) = C(sI - A)^{-1}B$$

Centro de Investigaciones Avanzadas en Ingeniería
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 25

Ley de control por realimentación del estado:

$$\dot{x}' = [A - BK]x$$

$$y = Cx$$

Centro de Investigaciones Avanzadas en Ingeniería
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 26

Ramas de control lineal moderno más difundidas:

- Control robusto
- Control óptimo

Técnicas de diseño:

- Filtros Kalman
- LQR
- LQG
- Hoo
- Realimentación del estado

Centro de Investigaciones Avanzadas en Ingeniería
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 27

Otras :

- Control no lineal
- Control adaptativo

Técnicas de Diseño:

- Modos Deslizantes
- Modelo de Referencia
- Modelo Inverso
- Gain Scheduling

Escuela de Ingeniería y Arquitectura de la Universidad de Sevilla

PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 28

Control inteligente

Escuela de Ingeniería y Arquitectura de la Universidad de Sevilla

PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 29

Antecedente históricos

- Siglo 17: Gottfried Leibniz
- Siglo 19: Charles Babbage, Máquina Analítica
- En 1936: Alan Turing propuso la construcción de una máquina que pudiera pensar.
- En 1930: Se dio comienzo a la Cibernética.
- 50's: McCarthy, Minsky, Newell y Simon
Inteligencia Artificial (IA)
El Dr. K. S. Fu introduce por primera vez el término "control inteligente"
- Integración de la IA con avanzados algoritmos de control.

Escuela de Ingeniería y Arquitectura de la Universidad de Sevilla

PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 30

Definición y Aspectos Básicos

- Inteligencia : el grado de autonomía
- Autonomía : auto-independencia
- Sistema de Control Inteligente

Un sistema es inteligente si tiene la habilidad de actuar apropiadamente en un ambiente incierto, donde una acción apropiada es aquella que aumenta la probabilidad de éxito y donde éxito es el logro de sub-metas de comportamiento que llevan al logro de la meta última del sistema.

Caracteres del Laboratorio de Automatización Dinámica
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 31

Aspectos generales de Controladores Inteligentes

- *Adaptación y Aprendizaje
- *Autonomía
- *Arquitectura funcional
- *Ambientes variables e inciertos
- *Estructuras de control jerárquico y/o distribuido
- *Sistemas Autónomos
- *Interacción con sistemas biológicos

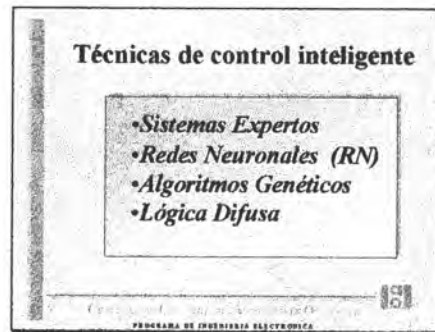
Caracteres del Laboratorio de Automatización Dinámica
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 32

| Control Inteligente | Control Convencional |
|--|---|
| <ul style="list-style-type: none"> •Expresiones Cualitativas •Basados en la experiencia: diseñador operario •No lineal •Útiles con datos ruidosos y en ambientes inciertos •Autonomía | <ul style="list-style-type: none"> •Modelos matemáticos lineales •Permite análisis: <ul style="list-style-type: none"> Estabilidad Robustez Controlabilidad Observabilidad |

Caracteres del Laboratorio de Automatización Dinámica
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 33



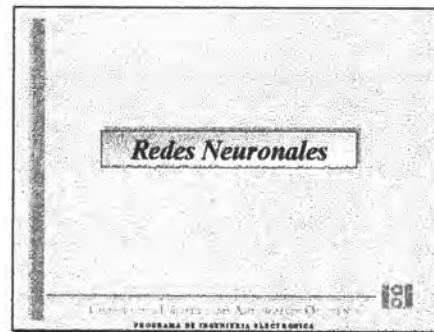
Diapositiva 34



Diapositiva 35



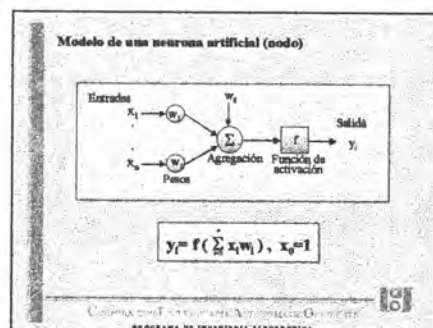
Diapositiva 36



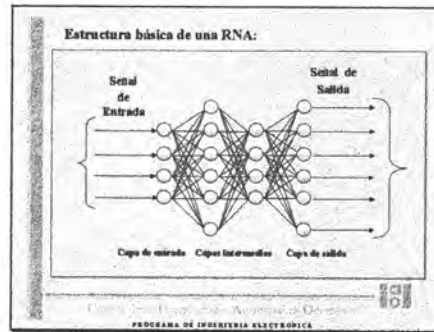
Diapositiva 37



Diapositiva 38



Diapositiva 39



Diapositiva 40

Entrenamiento mediante Backpropagation:

$E_p = \sum_k (d_k - x_k)^2$ d_k , salida deseada y x_k salida actual para el nodo k

$$\varepsilon_y = \frac{\partial^* E_p}{\partial x_y} = \frac{\partial E_p}{\partial x_y} = -2(d_k - x_k)$$

$$\varepsilon_x = \frac{\partial^* E_p}{\partial x_x} = \frac{\partial^* E_p}{\partial x_y} \frac{\partial f_y}{\partial x_x} + \frac{\partial^* E_p}{\partial x_z} \frac{\partial f_z}{\partial x_x} = \varepsilon_y \frac{\partial f_y}{\partial x_x} + \varepsilon_z \frac{\partial f_z}{\partial x_x}$$

Escuela Superior Politécnica de Ingeniería de Oroya
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 41

Aplicaciones en control

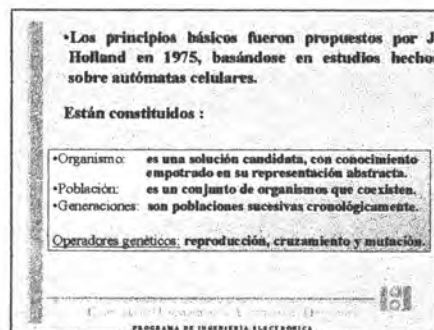
- Identificación de patrones (para identificar sistemas, clasificar condiciones, procesamiento de imágenes, etc.)
- Aprendizaje de modelos dinámicos directos (simulación de plantas) o inversos (control en cascada).
- Como mecanismo de adaptación para controladores convencionales.

Escuela Superior Politécnica de Ingeniería de Oroya
PROGRAMA DE INGENIERÍA ELÉCTRICA

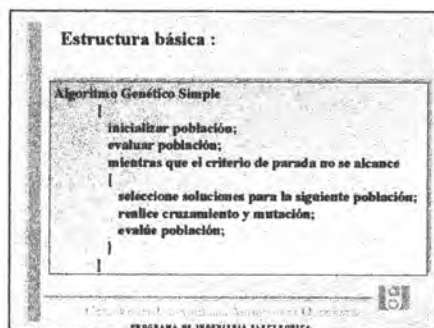
Diapositiva 42



Diapositiva 43



Diapositiva 44



Diapositiva 45

Aplicaciones:

- Identificación de parámetros y de sistemas
- Control robusto
- Sistemas de navegación autónomos
- Reconocimiento de patrones y de voz
- Chips bajo la tecnología VLSI
- Optimización

Escuela de Ingeniería Electrónica, Universidad de Buenos Aires

PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 46

Lógica difusa

Escuela de Ingeniería Electrónica, Universidad de Buenos Aires

PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 47

- Fue introducida por Lotfi Zadeh como la teoría de "Conjuntos Difusos" en 1965.
- Los conjuntos difusos son una generalización de la teoría de conjuntos convencional y contienen objetos que pertenecen imprecisamente al conjunto.
- Esta técnica permite una forma de representar las incertidumbres de un modelo matemático.
- Fue aplicada por primera vez para aspectos de control por Mamdani en 1974.

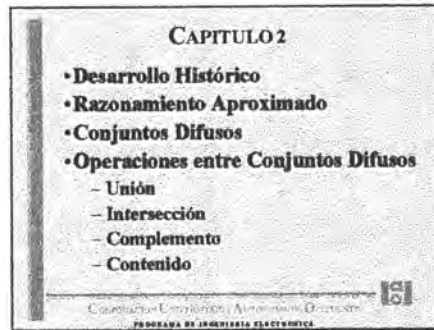
Lotfi Zadeh



Escuela de Ingeniería Electrónica, Universidad de Buenos Aires

PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 48



Diapositiva 49



Diapositiva 50



Diapositiva 51



1975 Desarrolla el concepto de Variable Lingüística y le da forma definitiva a la Regla Composicional de Inferencia.

1978 Publica un artículo sobre la teoría de posibilidades y la Lógica Difusa.

1979 En varios artículos describe una elaborada teoría sobre el Razonamiento Aproximado.

Lotfi Zadeh
Padre de la lógica Fuzzy

CONVENIO DE COLABORACIÓN ACADÉMICA DE GARCÍA
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 52

Razonamiento Aproximado

- Esta teoría permite las construcciones lógicas usadas para crear y manipular sistemas difusos
- Es una teoría mas general que la lógica difusa para el cálculo de sentencias difusas.



```

graph TD
    A[Razonamiento Aproximado] --> B[Lógica Difusa]
    B --> C[Teoría de Conjuntos Difusos]
  
```

CONVENIO DE COLABORACIÓN ACADÉMICA DE GARCÍA
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 53

CONJUNTOS DIFUSOS

CONVENIO DE COLABORACIÓN ACADÉMICA DE GARCÍA
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 54

Definición: Según Zadeh: "Un conjunto difuso es una clase de objetos con un continuo grado de pertenencia. Tal conjunto está caracterizado por una función de pertenencia (característica) la cual asigna a cada objeto un valor de pertenencia entre cero y uno".

• Un conjunto difuso A en U puede ser representado por :

$$A = \{(x, \mu_A(x)) | x \in U\}$$

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE GUATEMALA
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 55

• Un conjunto difuso puede verse como una generalización del concepto de un conjunto clásico cuya función de pertenencia sólo puede tomar uno de dos valores $\{0,1\}$.

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE GUATEMALA
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 56

• Cuando U es continuo, el conjunto difuso A se puede escribir como :

$$\int_U \mu_A(x) / x$$

• En el caso de que el universo de discurso sea discreto A se puede representar como

$$A = \sum_{i=1}^n \mu_A(x_i) / x_i$$

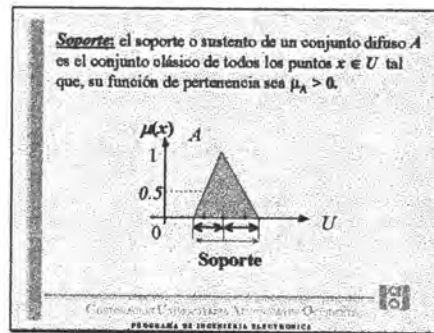
• Toda la información de un conjunto difuso está contenido en su función de pertenencia $\mu(x)$.

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE GUATEMALA
PROGRAMA DE INGENIERÍA ELÉCTRICA

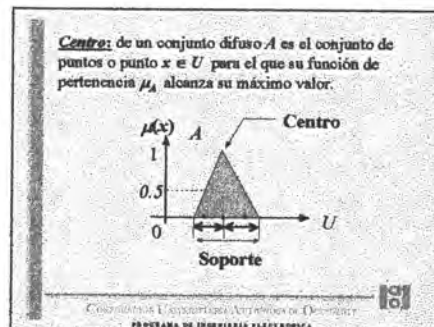
Diapositiva 57



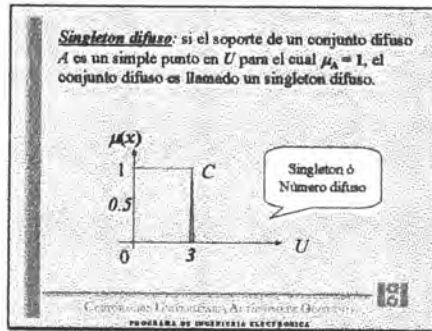
Diapositiva 58



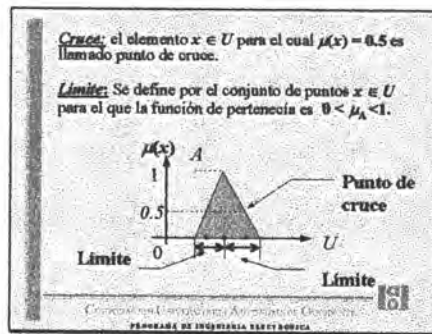
Diapositiva 59



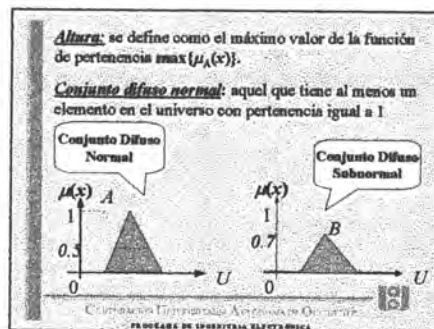
Diapositiva 60



Diapositiva 61



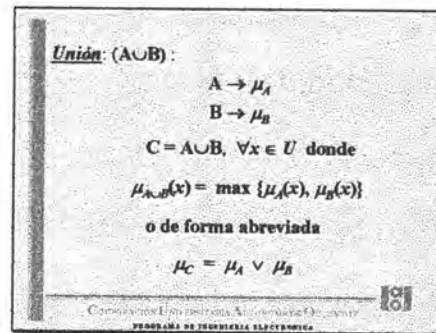
Diapositiva 62



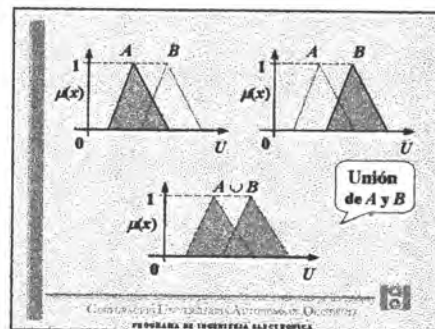
Diapositiva 63



Diapositiva 64



Diapositiva 65



Diapositiva 66

La intersección: $(A \cap B)$

$C = A \cap B \quad \forall x \in U \text{ donde}$

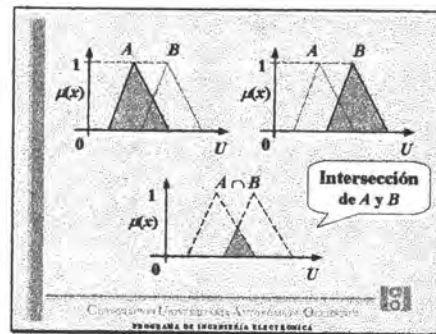
$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$$

o de forma abreviada

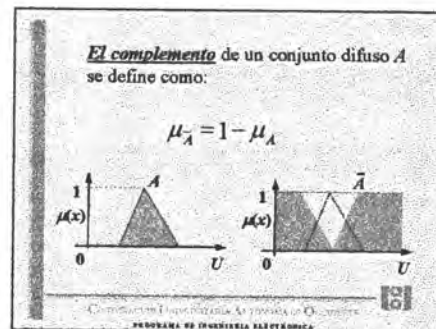
$$\mu_C = \mu_A \wedge \mu_B$$

CONICIA - Universidad Argentina de Occidente
PROGRAMA DE INGENIERÍA ELÉCTRICA

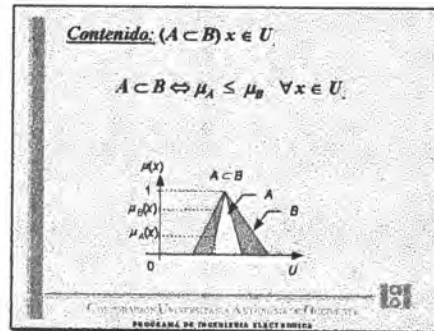
Diapositiva 67



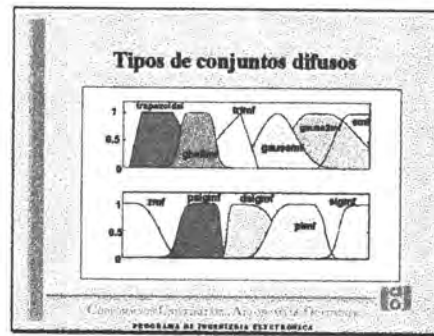
Diapositiva 68



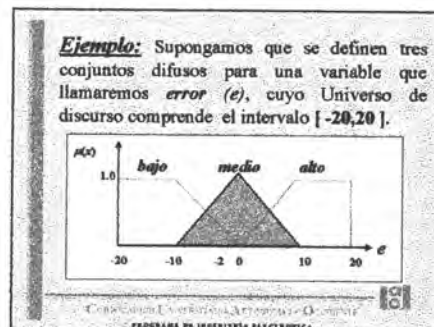
Diapositiva 69



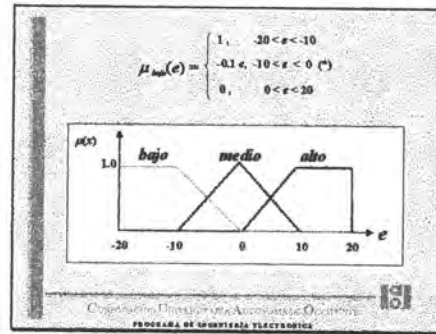
Diapositiva 70



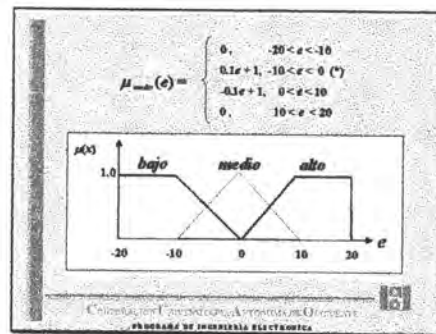
Diapositiva 71



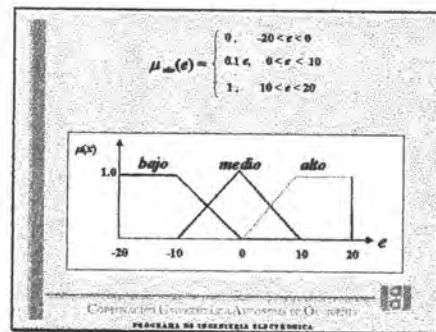
Diapositiva 72



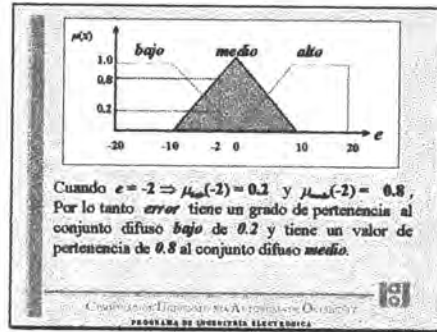
Diapositiva 73



Diapositiva 74



Diapositiva 75



Diapositiva 76

CAPITULO 3

- Variable lingüística
- Reglas IF-THEN difusas
- Fuzzificación
- Base de reglas y motor de inferencia
- Defuzzificación
- Sistemas difusos
 - Sistemas de lógica difusa puros
 - Sistemas Takagi y Sugeno
 - Sistemas de lógica difusa con fuzzificación y defuzzificación (tipo Mamdani)
- Aplicaciones

181

Universidad Autónoma de Occidente
PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 77

Variable Lingüística

181

Universidad Autónoma de Occidente
PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 78

• **Definición formal:** Una variable lingüística está caracterizada por una tripleta $(x, T(x), U)$ en donde x es el nombre de la variable lingüística; $T(x)$ es el conjunto de términos de x , que son, el conjunto de nombres de valores lingüísticos de x con cada valor dentro de un conjunto difuso universal o Universo de Discurso U .

181

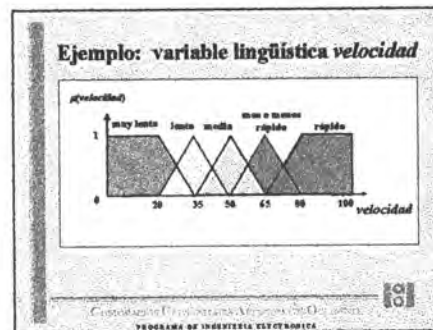
Universidad Autónoma de Occidente
PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 79

•**Definición intuitiva:** Si una variable puede tomar palabras en lenguaje natural (por ejemplo, pequeño, rápido, etc.) como sus valores, esta variable se puede definir como una variable lingüística.

Copyright 1997, Universidad Argentina de Informática
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 80



Diapositiva 81

Reglas IF-THEN difusas

Copyright 1997, Universidad Argentina de Informática
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 82

Reglas If-Then Difusas:

- Son proposiciones simples o compuestas.
- Están compuestas por una premisa y una conclusión o consecuente.

Estructura :

If <premisa> then <conclusión>

Universidad Tecnológica de Oaxaca
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 83

- Varias proposiciones simples difusas se conjugan por medio de un operador conjunción: **AND**.

Por ejemplo:

If < x es A and y es B > then <conclusión>

Universidad Tecnológica de Oaxaca
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 84

Fuzzificación

Universidad Tecnológica de Oaxaca
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 85

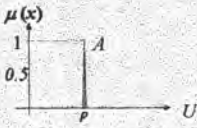
La **Fuzzificación** realiza un mapeo desde un punto crisp $\underline{x} = (x_1, \dots, x_n)^T \in U$ a un conjunto difuso A en U . Hay (por lo menos) dos opciones posibles de este mapeo.

- Singleton
- No Singleton

Comunicación y Fuzzificación en Sistemas de Fuzzificación
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 86

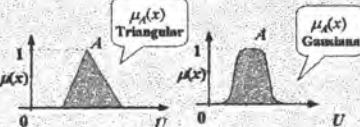
• **Fuzzificación Singleton:** A es un singleton difuso para los x , tales que para un punto $p \in U$, su pertenencia se define como $\mu_A(p) = 1$ y $\mu_A(x) = 0$ para todos los otros $x \in U$ con $x \neq p$.



Comunicación y Fuzzificación en Sistemas de Fuzzificación
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 87

• **Fuzzificación Nonsingleton:** A es un nonsingleton difuso si la función de pertenencia que lo caracteriza es de forma triangular, gaussiana, trapezoidal, etc.



Comunicación y Fuzzificación en Sistemas de Fuzzificación
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 88

Base de reglas y motor de inferencia

Universidad Autónoma de Occidente
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 89

Base de reglas: Colección de sentencias IF-THEN:

R_1 : IF x es A_1 and y es B_1 , THEN z es C_1
 R_2 : IF x es A_2 and y es B_2 , THEN z es C_2
 R_n : IF x es A_n and y es B_n , THEN z es C_n

Universidad Autónoma de Occidente
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 90

- La base de reglas es el corazón del sistema de lógica difusa: los otros tres componentes (fuzzificación, defuzzificación y motor de inferencia) son usados para interpretar estas reglas y hacerlas útiles para problemas específicos.
- Las reglas difusas IF-THEN proveen una estructura muy conveniente para que expertos humanos puedan expresar su dominio del conocimiento.

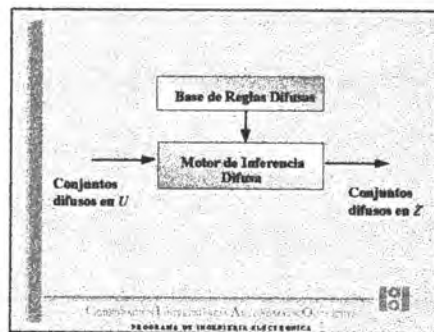
Universidad Autónoma de Occidente
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 91

Motor de inferencia: en él los principios de lógica difusa son usados para combinar las reglas difusas IF-THEN de la base de reglas en un mapeo desde conjuntos difusos en $U = U_1 \times \dots \times U_n$ a conjuntos difusos en Z .

Universidad Tecnológica de México
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 92



Diapositiva 93

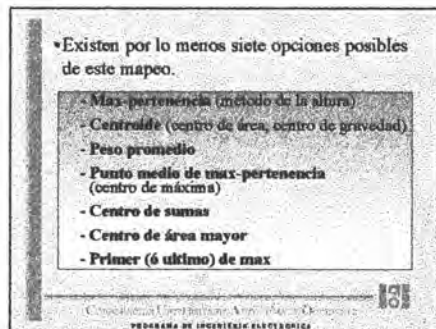
Defuzzificación

Universidad Tecnológica de México
PROGRAMA DE INGENIERÍA ELÉCTRICA

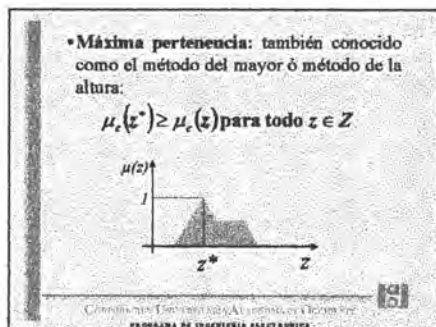
Diapositiva 94



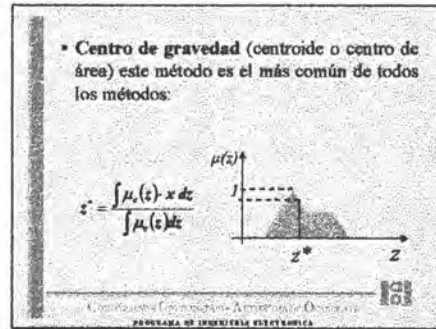
Diapositiva 95



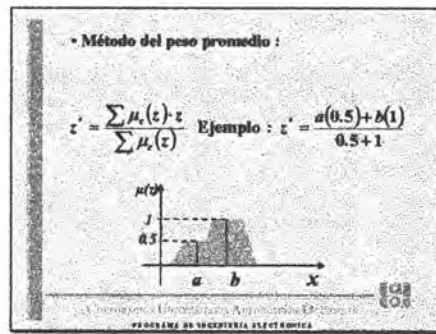
Diapositiva 96



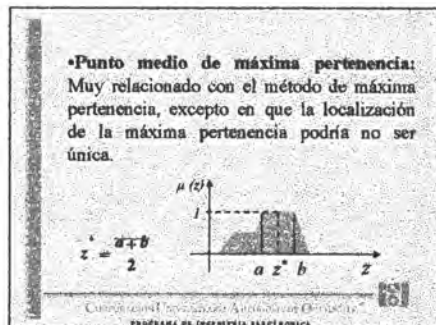
Diapositiva 97



Diapositiva 98

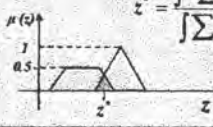


Diapositiva 99



Diapositiva 100

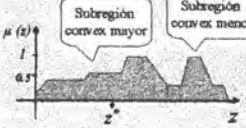
• **Centro de sumas:** este método es el más rápido de todos los métodos de defuzzificación. Involucra la suma algebraica de los conjuntos difusos de salida.

$$z^* = \frac{\int z \sum_{k=1}^n \mu_{C_k}(z) dz}{\int \sum_{k=1}^n \mu_{C_k}(z) dz}$$


Comisión Universitaria Argentina de Océanos
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 101

• **Centro de área mayor:** si el conjunto difuso de salida tiene al menos dos subregiones convexas, entonces z^* se calcula usando el método del centroide de la subregión de área mayor.

$$z^* = \frac{\int \mu_c(z) \cdot z \, dz}{\int \mu_c(z) dz}$$


Comisión Universitaria Argentina de Océanos
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 102

• **Primer (ó último) de máxima:** usa la unión de todos los conjuntos de salida C_k para determinar el valor más pequeño del dominio con máximo grado de pertenencia en C_k .

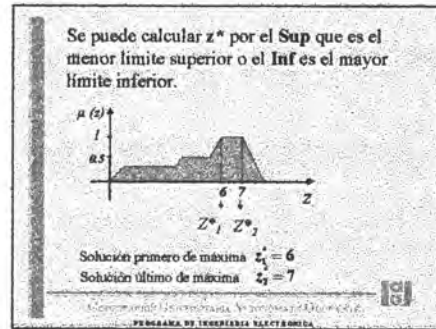
$$Altura(C_k) = \sup_{m \in Z} \mu_{C_k}(z)$$

$$z^* = \inf_{m \in Z} \{z \in Z \mid \mu_{C_k}(z) = Altura(C_k)\}$$

$$z^* = \sup_{m \in Z} \{z \in Z \mid \mu_{C_k}(z) = Altura(C_k)\}$$

Comisión Universitaria Argentina de Océanos
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 103



Diapositiva 104



Diapositiva 105

•Es el nombre dado a los sistemas que tienen una relación directa con conceptos difusos (como conjuntos difusos, variables lingüística, etc.) y lógica difusa.

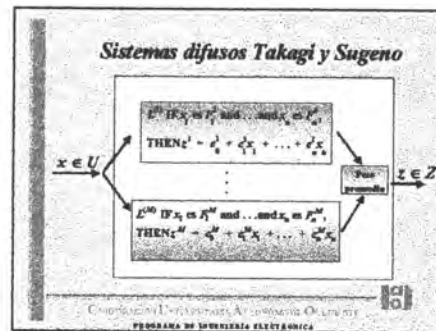
- Puros
- Takagi y Sugeno
- Mamdani.

Unidad de Investigación y Desarrollo en Ingeniería de Control y Robótica
 PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 106



Diapositiva 107



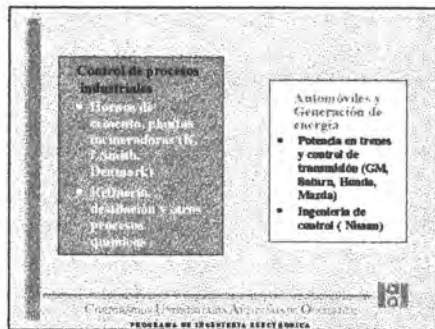
Diapositiva 108



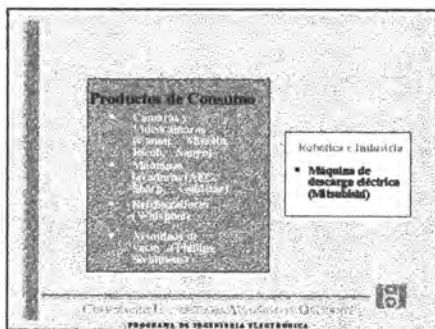
Diapositiva 109



Diapositiva 110



Diapositiva 111



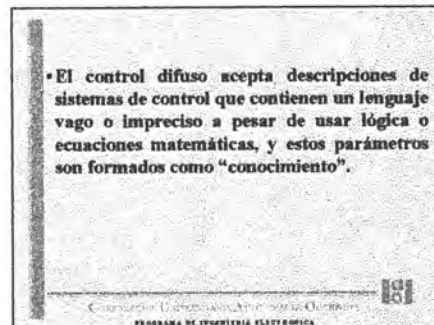
Diapositiva 112



Diapositiva 113



Diapositiva 114



Diapositiva 115

• Usando los datos de entrada obtenidos con base en un objetivo de control, la inferencia difusa puede determinar el mejor dato de salida para el control del sistema, permitiendo así implementar una estrategia de control lingüística basada en reglas que simula los mecanismos de decisión humana.

Centro de Investigación y Desarrollo en Ingeniería de Control y Automatización
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 116

Desarrollo histórico

Centro de Investigación y Desarrollo en Ingeniería de Control y Automatización
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 117

| Año | Autor | Contribución |
|------|---------------------------|---|
| 1972 | Zadeh | Base teórica para el control difuso |
| 1973 | Zadeh | Enfoque lingüístico |
| 1974 | Mamdani & Assilian | Control de una máquina de vapor |
| 1976 | Ross & Rutherford | Análisis de algoritmos de control |
| 1977 | Ostergaard | Control de sistemas con cambios de calor y en modo de control |
| 1977 | Wibaczyni et al. | Control difuso aplicado |
| 1979 | Kolomoiev et al. | Automata finito |
| 1980 | Hitachi ATO | Operación automática de su tren |
| 1980 | Tong et al. | Control del proceso de agua residual |
| 1980 | Fukami, Mizumoto y Tanaka | Inferencia condicional borrosa |
| 1983 | Hirota & Psatyris | Conjuntos difusos probabilísticos |
| 1983 | Takagi & Sugeno | Definición de reglas de control difuso |
| 1983 | Yasumoto, Miyamoto et al. | Control difuso predictivo |

Centro de Investigación y Desarrollo en Ingeniería de Control y Automatización
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 118

| Año | Autor | Contribución |
|------|--------------------------|--|
| 1964 | YIP-Hsi (Taiwan) | Se establece la IFSA |
| 1964 | Sugeno & Mitrakani | Control de potencia de un motor |
| 1965 | Klaska, Gupta et al. | Estabilización de potencia difusa |
| 1965 | Togai & Watanabe | Chip difuso |
| 1966 | Yamakawa | Microprocesador en controlador difuso |
| 1967 | Boon Difuso | Producción de consumo en Japón |
| 1967 | Sanday (Japón) | Operación de tres subestaciones de energía |
| 1968 | Dubois & Prade | Razonamiento aproximado |
| 1969 | MITI (Japón) | SI (sistema) LIPS |
| 1969 | IFSA (Japón) | Se crea SCFT |
| 1990 | Lee | Lógica difusa en control de motores |
| 1990 | Mitsubishi Electric Ind. | Control difuso de una lavadora |
| 1991 | Mitsubishi Electric C. | Video conferencia utilizando inferencia difusa |

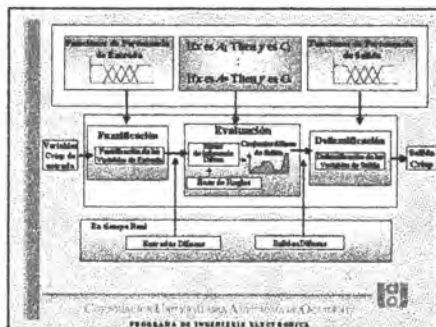
Nota: Mitsubishi Electric Industrial Co. Ltd. se el nombre dado a Panasonic en Japón

Contribución Universidad Autónoma de Occidente
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 119



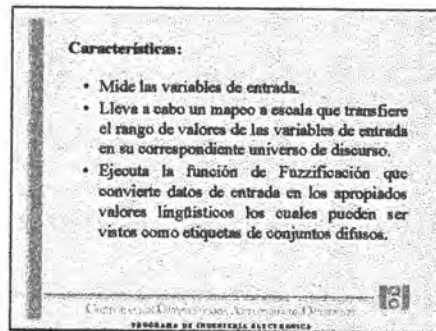
Diapositiva 120



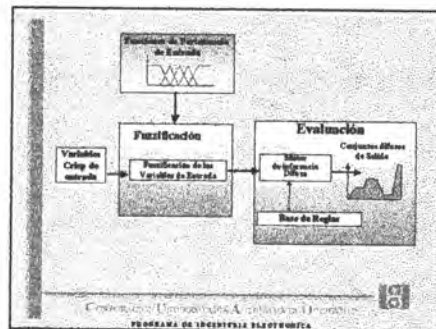
Diapositiva 121



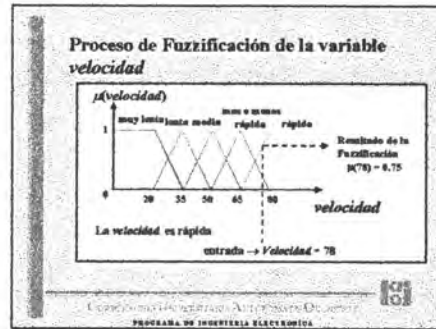
Diapositiva 122



Diapositiva 123



Diapositiva 124



Diapositiva 125



Diapositiva 126

Comprende un conocimiento del dominio de la aplicación y el servicio del objetivo de control. La cual involucra una "base de datos" y una "base de reglas de control lingüísticas (difusa)."

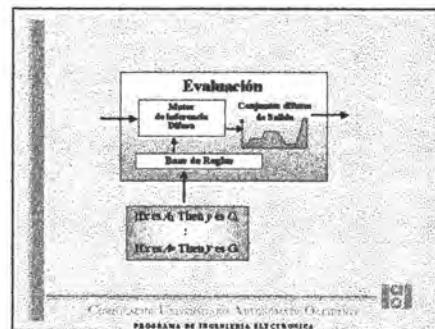
UNIVERSIDAD DE LA GUAYANA
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 127

- La base de datos proporciona necesariamente definiciones, que son usadas para definir reglas de control lingüísticas y manipulación de datos difusos en un CD.
- La base de reglas caracteriza el objetivo de control y la política de control del dominio del experto por medio de un conjunto de reglas de control lingüísticas.

Universidad Autónoma de Querétaro
 PROGRAMA DE INGENIERÍA ELECTRONICA

Diapositiva 128



Diapositiva 129

- Con la base de reglas se representa de manera estructurada la estrategia de control lingüística del operador experto, como un conjunto de reglas If-Then

If (estado del proceso) Then (salida de control)

Universidad Autónoma de Querétaro
 PROGRAMA DE INGENIERÍA ELECTRONICA

Diapositiva 130

• **Agregación de reglas difusas:** es el proceso para obtener el consecuente completo (conclusión) a partir de la contribución individual de los consecuentes de cada regla en la base de conocimiento.

Estrategias de agregación

- Conjunción
- Disyunción

Universidad Veracruzana - Apizaco - Oaxaca
PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 131

Conjunción: es este caso la agregación de salida (y) se obtiene por la intersección difusa de todos consecuentes de las reglas individuales, y^i , con $i = 1, 2, \dots, n$.

$$y = y^1 \text{ and } y^2 \text{ and } \dots \text{ and } y^n$$

$$y = y^1 \cap y^2 \cap \dots \cap y^n$$

para funciones de pertenencia

$$\mu_y(y) = \min(\mu_{y^1}(y), \mu_{y^2}(y), \dots, \mu_{y^n}(y)) \text{ para } y \in Y$$

Universidad Veracruzana - Apizaco - Oaxaca
PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 132

Disyunción: es este caso la agregación de salida (y) se obtiene por la unión difusa de todos consecuentes de las reglas individuales, y^i , con $i = 1, 2, \dots, n$.

$$y = y^1 \text{ or } y^2 \text{ or } \dots \text{ or } y^n$$

$$y = y^1 \cup y^2 \cup \dots \cup y^n$$

para funciones de pertenencia

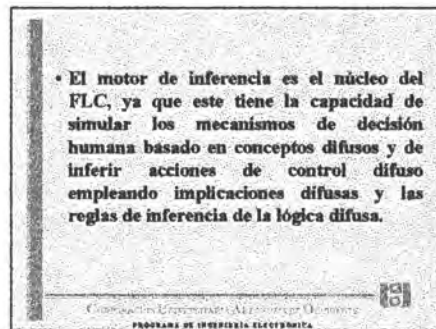
$$\mu_y(y) = \max(\mu_{y^1}(y), \mu_{y^2}(y), \dots, \mu_{y^n}(y)) \text{ para } y \in Y$$

Universidad Veracruzana - Apizaco - Oaxaca
PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 133



Diapositiva 134



Diapositiva 135



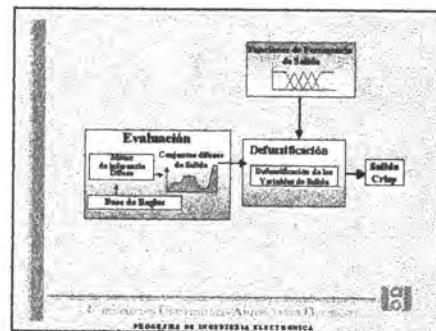
Diapositiva 136

- Convierte la información lingüística contenida en los conjuntos difusos de salida en valores crisp para que puedan ser interpretados por los actuadores (niveles de voltajes, presión o corriente, etc.)

136

Universidad Autónoma de Aguascalientes
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 137



Diapositiva 138

- Los mas importantes procedimientos de defuzzificación son los cinco siguientes:

- Centro de gravedad
- Centro de las sumas
- Centro de las mayores áreas
- Primer máximo
- Elección por la altura

138

Universidad Autónoma de Aguascalientes
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 139



Diapositiva 140

| <u>Aplicación</u> | <u>Descripción</u> |
|-------------------|--|
| •Cámaras | De entre diversos puntos de foco se escoge a través de reglas if-then el foco óptimo. |
| •Video Cámaras | En la estabilización de imágenes. |
| •Lavadoras | Mediante sensores se determina que tan sucia está el agua y cómo se va limpiando para determinar el ciclo de lavado apropiado. |
| •Fotocopias | Control electrográfico y calidad de impresión. |

LA UNIVERSIDAD ECUATORIANA
PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 141

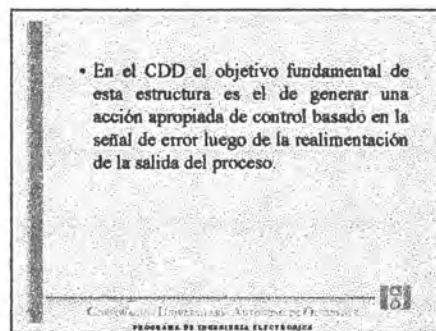
| <u>Aplicación</u> | <u>Descripción</u> |
|--------------------|---|
| •Automóviles | En el sistema de transmisión. |
| •Grúas Automáticas | En el cargue y descargue automático de containers en buques. |
| •Control de trenes | CSC Controlador de velocidad constante y TASC. Control automático de parada para el tren. |
| •Televisores | Control de la calidad de las imágenes. |

LA UNIVERSIDAD ECUATORIANA
PROGRAMA DE INGENIERIA ELECTRONICA

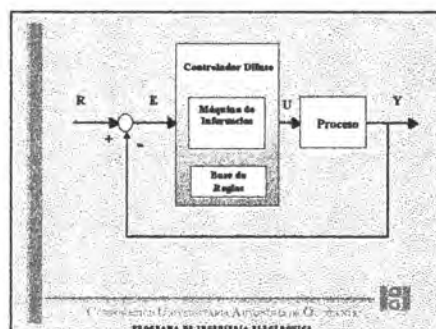
Diapositiva 142



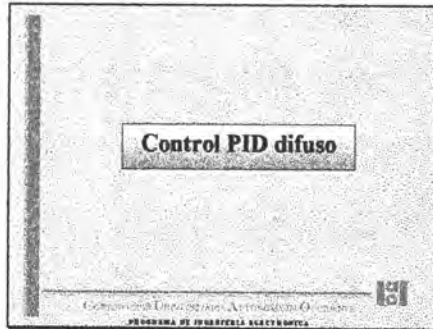
Diapositiva 143



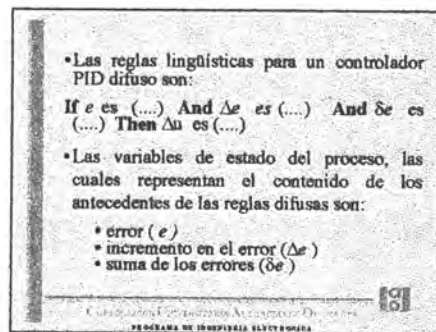
Diapositiva 144



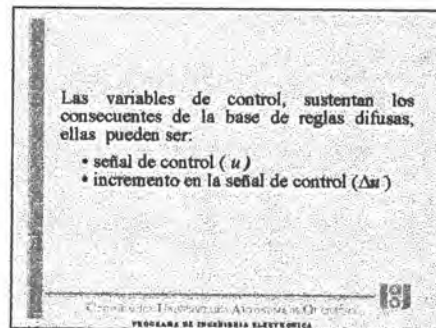
Diapositiva 145



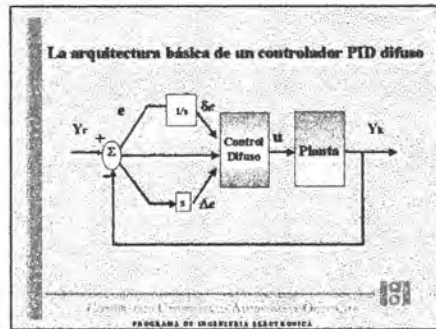
Diapositiva 146



Diapositiva 147



Diapositiva 148



Diapositiva 149

Elaboración de las reglas de control:

- Un controlador PID difuso con siete conjuntos difusos por variable, requiere de 343 reglas de control.
- Los controladores difusos PI y PD con igual cantidad de conjuntos difusos sólo requieren de 49 reglas de control.

Universidad de Cádiz
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 150

- Es por esto que es más fácil el diseño y la implementación de los PI y PD.
- Por simplicidad examinaremos las reglas para un controlador difuso PI, cuyas entradas son el error y el cambio en el error (e y Δe). Su salida será el incremento en la señal de control (Δu).

Universidad de Cádiz
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 151

De la misma manera que un controlador convencional discreto, se tiene que:

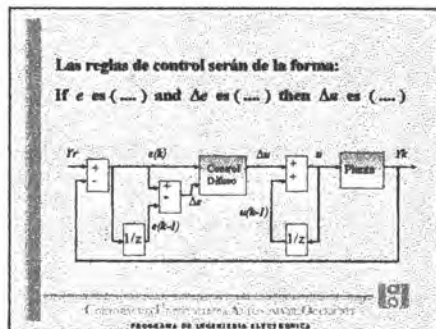
$$e(k) = Y_r(k) - Y(k)$$

$$\Delta e(k) = e(k) - e(k-1), \text{ de igual forma } \Delta u = u(k) - u(k-1)$$

$$\Delta e(k) = e(k) + e(k-1)$$

UNIVERSIDAD AUTÓNOMA DE COAHUILA
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 152



Diapositiva 153

Control Difuso por modos deslizantes

UNIVERSIDAD AUTÓNOMA DE COAHUILA
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 154

- Aplicado en sistemas altamente no lineales, variantes en el tiempo y con ambientes inciertos.
- Son un tipo particular de los sistemas de CEV, en los cuales la trayectoria del estado es obligada a dirigirse hacia una superficie dada y a "deslizar" por ella y de esta manera alcanzar el punto de operación deseado.

Comunicación, Electrónica y Automática - Opción
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 155

- Teóricamente, los elementos de conmutación del SCMD deberían operar a unas frecuencias infinitas para producir un modo deslizante ideal. Es por esto, que en la práctica se produce el "chattering" alrededor de la superficie de deslizamiento.

Comunicación, Electrónica y Automática - Opción
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 156

- Mediante la introducción de un sistema de control por modos deslizantes difuso (SCMDD) es posible contar con las bondades del controlador difuso y las del control por modos deslizantes.

Comunicación, Electrónica y Automática - Opción
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 157

• Fuzzificando la superficie de deslizamiento, el SCMD produce un modo deslizante difuso con mucho menos "chattering". El sistema en lazo cerrado, podrá estar caracterizado por una superficie de deslizamiento definida por el diseñador y las reglas difusas se pueden obtener fácilmente.

Controlador Controlado Automático Orientado
PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 158

• En la elaboración de las reglas lingüísticas de control, se debe tener en cuenta (en forma similar que con los SCMD) que por encima de la línea de conmutación, $S=0$, se genere una salida de control negativa y por debajo una positiva.

• Técnica general: partición del plano de fase

Controlador Controlado Automático Orientado
PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 159



Procedimiento de diseño:

• Es definido el error entre la salida del sistema $y(kT)$ y la señal de referencia $r(kT)$ como:

Controlador Controlado Automático Orientado
PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 160

$$e1(kT) = r(kT) - y(kT)$$

y $de1(kT) = e2(kT)$

La superficie de deslizamiento se define como

$$S = e2(kT) + \sigma e1(kT)$$

De igual manera $dS = \{ S(kT) - S((k-1)T) \} / T$

$$= \sigma e2(kT) - f(e1(kT)) - g(e1(kT))u(kT) - d(kT)$$

UNIVERSIDAD POLITÉCNICA AUTÓNOMA DE QUERÉTARO
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 161

Se asume que $g(e1(kT))$ es mayor que cero y $d(kT)$ es la señal de perturbación, por lo tanto realizando el producto SdS y haciendo que $S = 0$, se puede hallar el valor de σ .

Sean $V(x(t)) = \frac{1}{2} S^2$ y

$$dV(x(t)) = SdS$$

UNIVERSIDAD POLITÉCNICA AUTÓNOMA DE QUERÉTARO
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 162

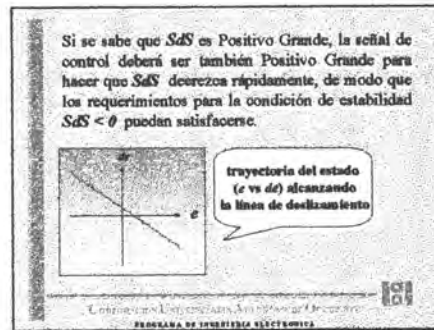
- Con el fin de satisfacer el teorema de estabilidad de Liapunov ($SdS < 0$), se definen el conjunto de reglas. Un ejemplo podría ser:

"Si S y dS son Positivo Grande Entonces u es Positivo Grande"

- Las reglas lingüísticas de control se pueden determinar intuitivamente mediante un análisis de la ecuación del producto SdS .

UNIVERSIDAD POLITÉCNICA AUTÓNOMA DE QUERÉTARO
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 163



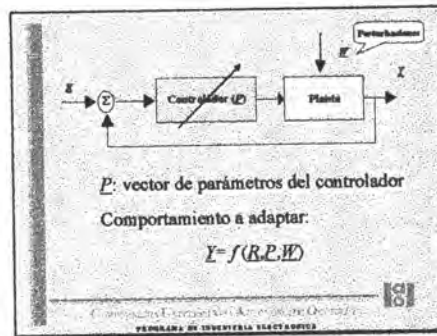
Diapositiva 164



Diapositiva 165

- Adaptación: aptitud de un sistema para acomodar (acostumbrar, adaptar) su comportamiento a cambios en el medio.
 - La dinámica de los sistemas hace necesario el cambio en los parámetros de los controladores ante cambios en el medio, de ahí surge la necesidad de controladores con capacidad de adaptación.
 - Ejemplo de adaptación es el ajuste de los parámetros de sintonía de un PID.
- Colombiana Universidad Autónoma de Bucaramanga
PROGRAMA DE INGENIERÍA ELÉCTRICA

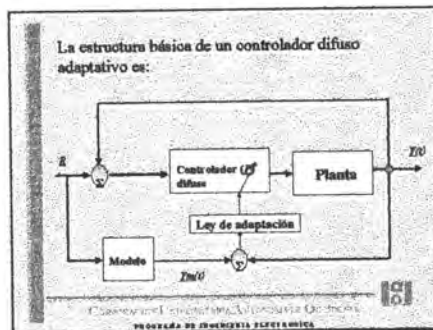
Diapositiva 166



Diapositiva 167

- En la implementación de CD existe la posibilidad de que la base de conocimiento requiera ajustar sus parámetros para implementar una correcta estrategia de control lingüística.
 - El primer controlador difuso adaptativo fue desarrollado en Queen Mary College por Procyk y Mamdani en 1979.
- PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 168



Diapositiva 169

- El modelo de referencia se usa para especificar la respuesta ideal que el controlador difuso debe conseguir.
- La ley de adaptación ajusta los parámetros del controlador difuso en línea de tal forma que la salida $\hat{Y}(t)$ siga la salida del modelo de referencia $Y_m(t)$.

Controlador Fuzzy Adaptativo On-line
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 170

Ventajas del control difuso adaptativo sobre el control difuso no adaptativo:

- Mejor desempeño debido a que se puede autoajustar ante los cambios de la planta.
- Requiere menos información acerca de la planta ya que la ley de adaptación ayuda a conocer la dinámica de la planta durante su operación.

Controlador Fuzzy Adaptativo On-line
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 171

Desventajas:

- El sistema de control resultante es más difícil de analizar porque no sólo es no lineal, sino que también es variante en el tiempo.
- Su implementación es más costosa.

Controlador Fuzzy Adaptativo On-line
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 172

Los parámetros (P) del controlador difuso que se pueden ajustar mediante la ley de control son:

- ☐ Los factores de escala para las variables de salida y de entrada.
- ☐ Elementos relacionales contenidos en la matriz relacional.
- ☐ Las funciones de pertenencia asociadas con las variables de entrada y de salida.

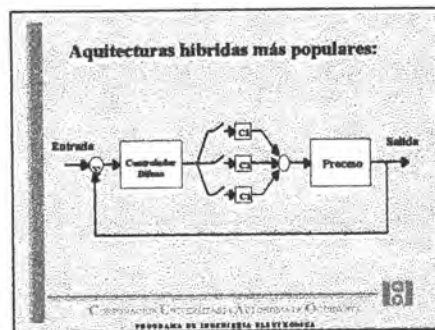
CONICET - Universidad Argentina de Ciencia y Tecnología
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 173

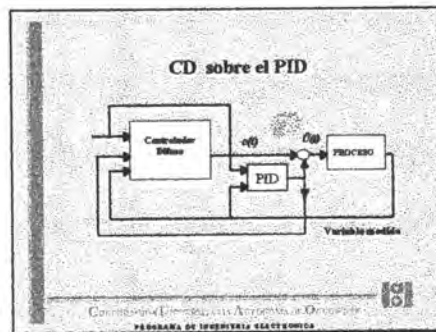
Otros enfoques

CONICET - Universidad Argentina de Ciencia y Tecnología
PROGRAMA DE INGENIERÍA ELÉCTRICA

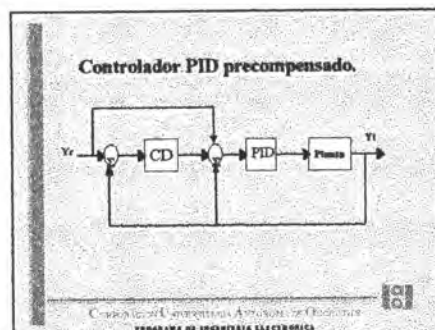
Diapositiva 174



Diapositiva 175



Diapositiva 176



Diapositiva 177



Diapositiva 178

CAPITULO V


•Estabilidad y robustez de controladores difusos



Centro de Investigación en Sistemas Inteligentes de Computación
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 179


- Actualmente no existe una teoría general sobre la estabilidad de los sistemas de control difusos.
- No se ha comprobado que una base de reglas de control, la cual sea considerada estable en su dominio difuso (donde la ley de control es lingüística por naturaleza o está dada en alguna forma numérica o funcional), satisfaga algún criterio analítico.



Centro de Investigación en Sistemas Inteligentes de Computación
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 180

- A pesar de la carencia de una metodología para el análisis de la estabilidad, desde el punto de vista práctico ha sido muy útil la prueba de prototipos. Para el caso del tren subterráneo de la ciudad de Sendai (Japón), Se realizaron cerca de **300.000** simulaciones para probar el sistema, el cual opera exitosamente desde 1987.



Centro de Investigación en Sistemas Inteligentes de Computación
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 181

•Existen muchos intentos para mostrar la estabilidad de los sistemas de control difuso, pero su aplicación está restringida a casos muy puntuales. Algunos de estos métodos son:

Centro de Investigación y Desarrollo en Ingeniería de Control y de Sistemas Automáticos
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 182

Estabilidad Energética :

Primero se define la energía de la salida del sistema difuso, para una entrada dada (X) y una relación difusa (R), se define la energía de salida del sistema difuso como: $E(Y) = E(X \circ R)$ donde Y es la salida difusa, E es la energía asociada con la función de pertenencia Y y (\circ) es el operador composicional de la regla de inferencia. Para chequear la estabilidad del sistema, se determinan los cambios en la energía de salida.

Por ejemplo $\Delta E = E(Y_k) - E(Y_{k-1})$, acorde con el teorema de estabilidad de Lyapunov, el sistema es estable si ΔE es negativo.

Centro de Investigación y Desarrollo en Ingeniería de Control y de Sistemas Automáticos
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 183

• Otra técnica utilizada es la de utilizar un *controlador supervisor* para las condiciones de inestabilidad, aquí el CD es diseñado primero sin tener ninguna consideración sobre la estabilidad, luego otro controlador es adicionado al CD para cuidar de los requerimientos de estabilidad. Debido a que lo que se requiere es que el CD realice la acción principal de control, el controlador adicional deberá ser un guardián en vez de un controlador principal.

Centro de Investigación y Desarrollo en Ingeniería de Control y de Sistemas Automáticos
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 184

- Por esto, se escoge al controlador adicional con las siguientes características: si el CD está trabajando bien, el controlador adicional permanecerá inactivo, si el sistema está tornándose inestable, el controlador adicional comienza su operación para garantizar la estabilidad. De esta manera el controlador adicional puede ser llamado un controlador supervisorio.

Universidad Tecnológica Autónoma de México
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 185

- Se han hecho muchos intentos para encontrar una teoría general sobre la estabilidad de los SCD. En 1992 fue publicado en la revista Fuzzy Sets and Systems Vol. 45, por los doctores Kazuo Tanaka y Michio Sugeno un artículo titulado "*Stability analysis & design of fuzzy control systems*", el cual es aplicado a los sistemas de control tipo Sugeno.

Universidad Tecnológica Autónoma de México
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 186

- Al aplicar la teoría de estabilidad de Lyapunov para un sistema discreto, es posible probar para un sistema difuso, con las condiciones dadas para sus funciones de pertenencia que el sistema es asintóticamente y globalmente estable.
- Otros estudios sobre la estabilidad de los SCD están basados en *métodos estadísticos*. Se propone el uso de ciertos índices de estabilidad, estos índices miden qué tan lejos/cerca está el sistema de perder la estabilidad.

Universidad Tecnológica Autónoma de México
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 187

- En cuanto a la robustez de los SCD se puede decir que tampoco existe una teoría bien elaborada y aceptada para realizar su análisis, lo que si se tiene reconocido es que la robustez de un SCD se encuentra implícita en sus reglas, ya que éstas están basadas en el conocimiento que un experto humano tiene del sistema en cuestión.
- Cada regla representa un comportamiento local, es decir cada regla describe una reacción ante una situación.

Universidad Nacional Autónoma de México
 Facultad de Ingeniería Electrónica

Diapositiva 188

- Si se encontraran problemas de estabilidad en esta situación, se puede modificar la regla respectiva, las demás reglas quedarían inalterables, dando paso a una optimización orientada hacia una meta.
- En un sistema de control convencional, cualquier modificación hecha para mejorar determinada situación puede producir efectos indeseados en el sistema bajo otras situaciones.

Universidad Nacional Autónoma de México
 Facultad de Ingeniería Electrónica

Diapositiva 192

Descripción del proceso:

El proceso consiste en la desulfuración del combustible entrante a la planta, el cual contiene un ácido de gas de petróleo licuado (ácido GPL) para producir gasolina y GPL en las siguientes etapas:

1. El combustible es mezclado con hidrógeno gaseoso.
2. La mezcla es separada en el ácido de gas y en combustible (nafta y LPG) dentro de un columnador.
3. El combustible (nafta y GPL) es enviado al separador en donde se divide en gasolina y en GPL, este último es extraído de la parte superior del separador y la gasolina de la parte inferior.

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 193

Problema de Control:

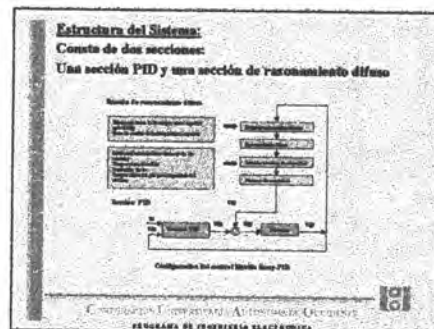
Cuando la temperatura en el calentador es muy alta, la calidad del GPL es muy pobre y cuando la temperatura es muy baja, la calidad de la gasolina decrece.

Es por esto que la temperatura del calentador debe ser mantenida en un punto fijo.

El sistema de control híbrido resuelve este problema al controlar la temperatura de la parte superior del separador.

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 194



Diapositiva 195

Un valor preliminar de la temperatura superior $u(t)$ es calculado por el PID basándose en el error de lazo cerrado $e(t) = T_r - T(t)$. El PID tiene la forma estándar:

$$u(t) = K_c e(t) + T_d \frac{de(t)}{dt} + \frac{1}{T_i} \int e(t) dt$$

El valor preliminar de salida $u(t)$ es compensado por la salida $z(t)$ de la sección de racoreamiento difuso, en el cual sus salidas corresponden a singleton. Las reglas difusas son de la forma:

If TT es A_i and ΔTT es B_j Then Y es c_k

TT : es $T(t) - T_r$, donde $T(t)$ es la temperatura superior del separador y T_r es la temperatura superior de referencia.

ΔTT : es la tasa de cambio de la temperatura superior del separador.

CONSEJO TECNOLÓGICO AUTÓNOMO DE OAXACA
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 196

Funciones de Pertenencia y Reglas de Control

Premisas:

Consecuentes:

Matriz de Reglas

| | ΔTT | | | | |
|------|-------------|----|----|----|----|
| TT | NB | NS | ZR | PS | PB |
| NB | P1 | P2 | P3 | P4 | P5 |
| NS | P2 | P3 | P4 | P5 | P6 |
| ZR | P3 | P4 | P5 | P6 | P7 |
| PS | P4 | P5 | P6 | P7 | P8 |
| PB | P5 | P6 | P7 | P8 | P9 |

CONSEJO TECNOLÓGICO AUTÓNOMO DE OAXACA
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 197

Resultados:

Este sistema de control se aplicó a una planta de producción de nafta hidro-desulfurada en la refinería de Minatitlán en Huixtla.

Operación con el controlador PID Difuso. Operación con el Control PID

El sistema es muy efectivo en el control de la planta actual, no solo en el estado estacionario, sino también durante el transitorio.

CONSEJO TECNOLÓGICO AUTÓNOMO DE OAXACA
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 198



Diapositiva 199



Diapositiva 200



Diapositiva 201

Durante la combustión el sistema de control debe mantener las siguientes condiciones:

- Control de la concentración de O_2 en el gas que sale de la cámara para mantenerlo sobre valores constantes.
- Mantener a la salida una uniformidad térmica.
- Mantener unas condiciones para un flujo óptimo entre la cámara y el primer quemador y así prevenir problemas con emisiones indeseables.

Colaboradora Universitat Autònoma de Barcelona
PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 202

Structure of the Control System

Estructura del sistema de control difuso

Proyecto: "Power Control"

Fuzzy Control Rules: Inactivation

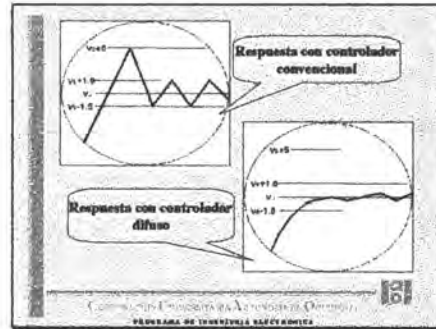
Colaboradora Universitat Autònoma de Barcelona
PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 203

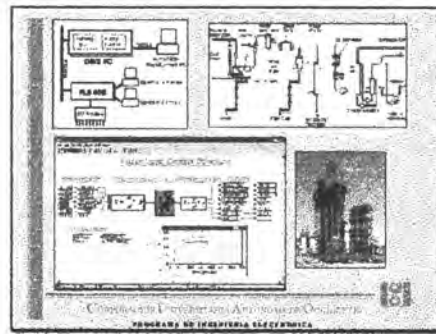
**LIMITADOR DE VELOCITAT
PARA CAMIONES DE CARGA**

Colaboradora Universitat Autònoma de Barcelona
PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 207



Diapositiva 208



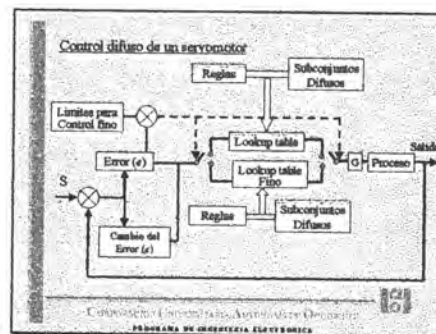
Diapositiva 209



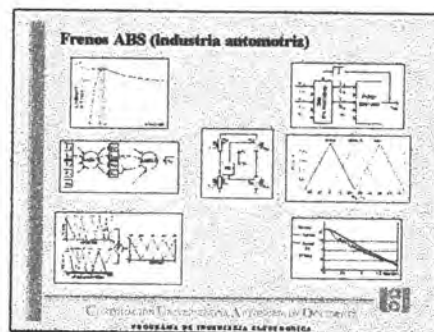
Diapositiva 210



Diapositiva 211



Diapositiva 212




Diapositiva 214

Capítulo VII

Herramientas de software para diseño, simulación e implementación de controladores difusos

- CUBICALC (v. 2.0)
- TILSHELL (v. 3.0)
- fuzzyTECH (v. 3.0 Online Edition)
- FIDE (FUZZY INFERENCE DEVELOPMENT ENVIRONMENT v. 2.0)
- Fuzzy Logic Toolbox para MatLab
- FUZZLE (v. 3.0)



Centro de Investigación y Desarrollo en Ingeniería de Control y de Sistemas
FACULTAD DE INGENIERÍA ELECTRÓNICA

Diapositiva 215

Herramientas de software




Centro de Investigación y Desarrollo en Ingeniería de Control y de Sistemas
FACULTAD DE INGENIERÍA ELECTRÓNICA

Diapositiva 216

- Diseñar
- Simular
- Implementar controladores difusos

• Permiten construir una base de reglas, definir funciones de pertenencia, analizar gráficamente la base de reglas y las salidas del sistema a implementar.

• Generar código fuente ó código ensamblador que implementa la base de reglas, y que puede ser llevado a un microcontrolador o un PLC para definir una estrategia de control difuso.



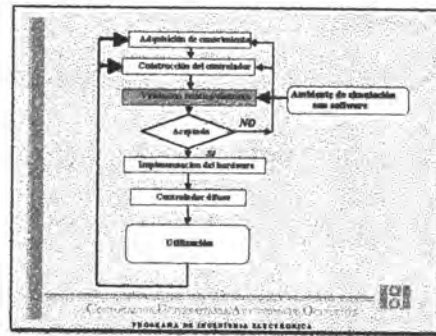
Centro de Investigación y Desarrollo en Ingeniería de Control y de Sistemas
FACULTAD DE INGENIERÍA ELECTRÓNICA

Diapositiva 217

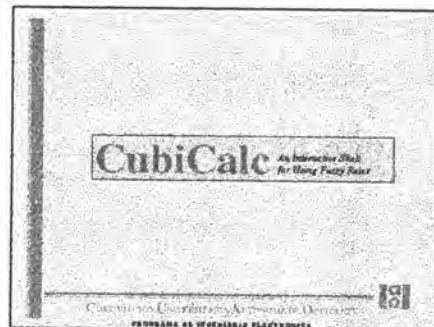
| Software | Empresa |
|------------------------------------|-------------------------|
| CubiCalc | HyperLogic Corp. |
| FUSHELL | Logic Infology |
| FuzzyTech | INFORM |
| FUSE | Apicom |
| Fuzzy | Integrated Systems Inc. |
| Fuzzy Knowledge Builder o Manifold | Fuzzy Systems |
| Graphical Editor | Signisec |
| Fuzzy C | Byte Craft Ltd. |
| Fuzzy Logic ToolBox para Matlab | Math Works Inc. |
| FS-10A I | Omron |
| FuzzySoft | OTS Travel OMBR |
| LFLC Plus 1.2 | Inst. de Genesys |
| FuzzyCrisp | Spilash Corp. |
| Vision ver. 1.5 | Vision Solutions Inc. |
| FuzzyCalc ver. 1.0 | FuzzyWare Inc. |
| KBC para MC68HC11 y MC68HC05 | Motorola Inc. |
| Fuzzy Logic Development Kit (FUDK) | TBI Enterprises Inc. |
| Fuzzy Logic Designer ver. 1.0 | Byte Dynamics Inc. |

Comunicación, Computación y Automatización de Procesos
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 218



Diapositiva 219



Diapositiva 220

| | |
|-------------------|--|
| Distribuidora: | <i>HyperLogic Corp.</i> |
| Versión: | <i>2.0</i> |
| Interfaz: | <i>GUIs</i> |
| Código Generador: | <i>Formato propio</i> |
| URL: | <i>http://www.hyperlogic.com</i> |



 Universidad Autónoma de Occidente

 PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 221

Requerimientos:

- IBM PC compatible 286 o superior
- Microsoft Windows 3.1 ó superior
- De 1 a 1.8 MB de espacio en disco duro (2.8 M RAM)
- Monitor VGA (recomendado)
- Mouse



 la solución más
 de Gran Precio Fácil




 Universidad Autónoma de Occidente

 PROGRAMA DE INGENIERÍA ELECTRÓNICA


Diapositiva 222

Características:

- Permite:
 - Ajustar los pesos de las reglas en tiempo real.
 - Amplio conjunto de funciones matemáticas.
 - Bajo costo (de mayor uso).
 - Los métodos de defuzzificación (Izquierda del máximo, derecha del máximo, mitad del máximo y centroide.)



 la solución más
 de Gran Precio Fácil



 Universidad Autónoma de Occidente

 PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 223

CUBICALC posee un editor de texto para ingresar las reglas como frases *if-then* y un editor gráfico para definir las funciones de pertenencia como una unión de líneas segmentadas. Usa pesos para las reglas, soporta modificadores lingüístico (hedges) en las reglas.

Ejemplo:
 (HRuleWeight) If (Temperature es algo caliente o la Humedad relativa) and
 Temperature cambia lo incrementado
 then Fuzzyto Zoharby por Alta
 @HRuleActv

Corporación Universitaria Autónoma Occidental
 PROGRAMA DE INGENIERÍA ELECTRONICA

Diapositiva 224

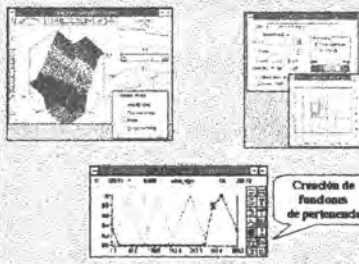
El peso de la regla puede ser especificado como una variable y ser usado iterativamente por el usuario o por otro módulo de computación. La frase opcional "*@HRuleActv*" asigna el nivel de activación para una variable llamada *HRuleActv*; esta variable puede entonces ser examinada para propósitos de depuración o por otro módulo de cálculo.



Hyperlogic

Corporación Universitaria Autónoma Occidental
 PROGRAMA DE INGENIERÍA ELECTRONICA

Diapositiva 225



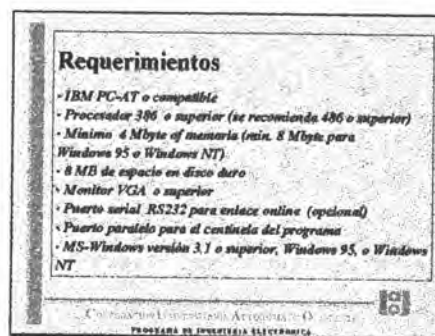
Creación de funciones de pertenencia

Corporación Universitaria Autónoma Occidental
 PROGRAMA DE INGENIERÍA ELECTRONICA

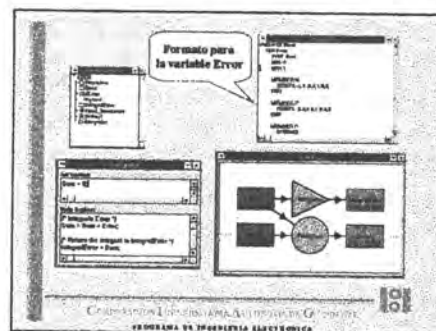
Diapositiva 229



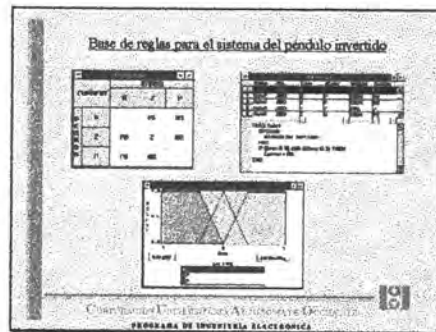
Diapositiva 230



Diapositiva 231



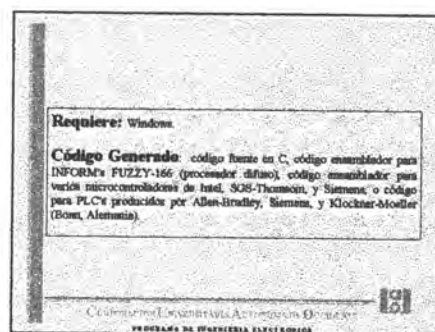
Diapositiva 232



Diapositiva 233



Diapositiva 234



Diapositiva 235

Ejemplo de un bloque de reglas:

```

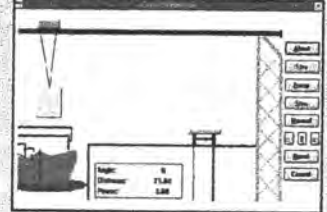
RULEBLOCK (
  INPUT = A, B; /* variables de entrada y salida para el bloque de reglas */
  OUTPUT = C;
  /* una regla es: operando con un valor de pertenencia 0 para
  asignarles la primera variable de verdad */
  AGREGACION = (MIN, MAX, PRODUCT);
  /* una segunda operando con un valor de pertenencia 0 para combinar
  valores de pertenencia de reglas con pesos de reglas */
  COMPOSICION = (SUMA, PROMEDIO);
  /* reglas activadas en este bloque */
  RULES (
    IF A = bajo AND B = alto THEN C = medio;
    WITH 0.8; /* esta regla tiene peso de 0.8 */
    IF A = alto AND B = alto THEN C = pequeño;
    WITH 0.9; /* esta regla tiene peso de 0.9 */
  )
)

```

Universidad Autónoma de Querétaro
FACULTAD DE INGENIERÍA ELECTRONICA

Diapositiva 236

Ejemplo del control de una grúa de contenedor de la versión 4.0 (demo)



| | |
|-----------|-------|
| Velocidad | 0 |
| Distancia | 15.00 |
| Peso | 2.00 |

Universidad Autónoma de Querétaro
FACULTAD DE INGENIERÍA ELECTRONICA

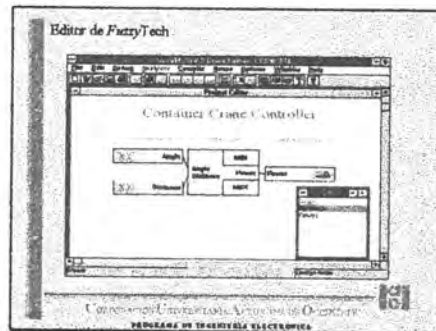
Diapositiva 237

Pantalla del control de una grúa de contenedor de la versión 4.0 (demo)

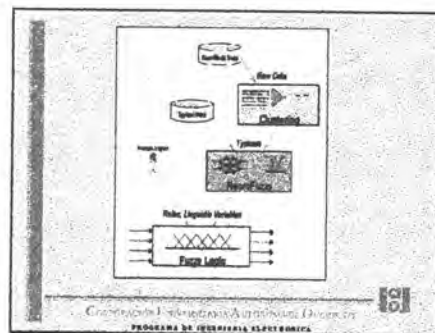


Universidad Autónoma de Querétaro
FACULTAD DE INGENIERÍA ELECTRONICA

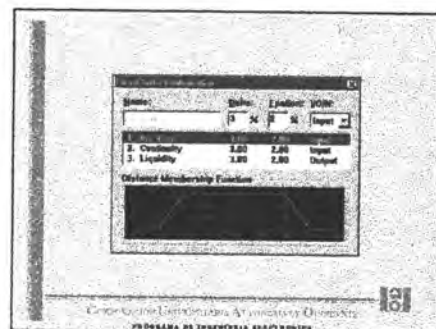
Diapositiva 238



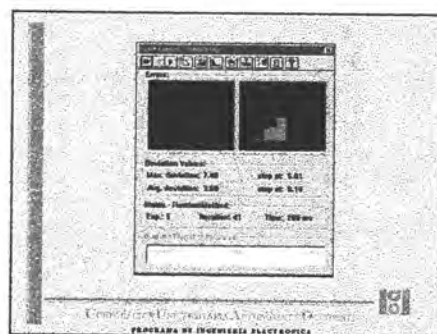
Diapositiva 239



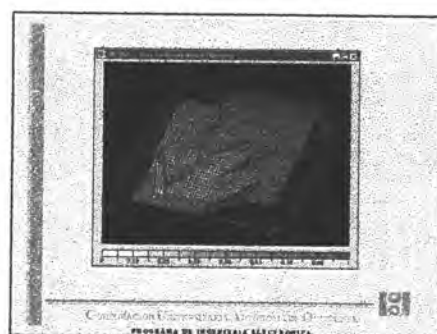
Diapositiva 240



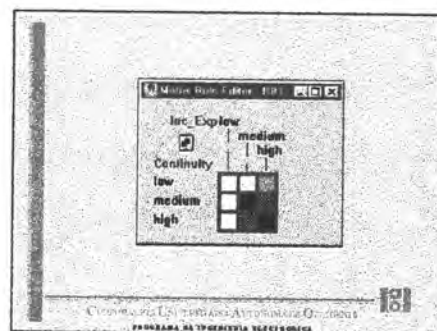
Diapositiva 241



Diapositiva 242



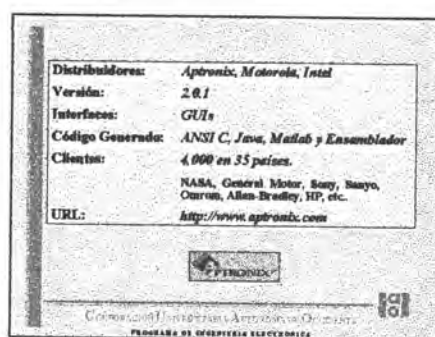
Diapositiva 243



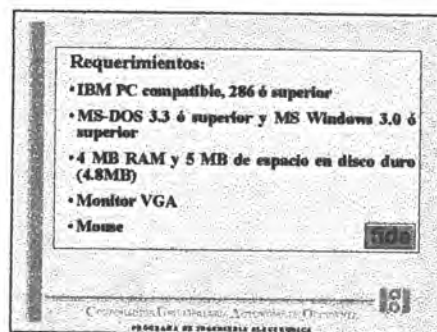
Diapositiva 244



Diapositiva 245



Diapositiva 246



Diapositiva 247

Características:

- El paquete viene con un disquete de 3½" y tres manuales.
- Permite:
 - 256 etiquetas por variable.
 - 256 variables.
 - 8100 reglas por unidad de inferencia (FIU)
 - 16.320 etiquetas por FIU

COMUNICACIÓN Y SISTEMAS AUTOMÁTICOS (COSA) - U. de Sevilla
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 248

- Inferencia: Mandani ó TVFI
- Operadores: min, max, producto, sum, intersección y unión.
- Fuzzificación: centralde, rightmost, rightmost y promedio maximizers
- Edición gráfica de las MF
 - Triangular
 - Polígono
 - Cúbica
 - Trazo libre

COMUNICACIÓN Y SISTEMAS AUTOMÁTICOS (COSA) - U. de Sevilla
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 249

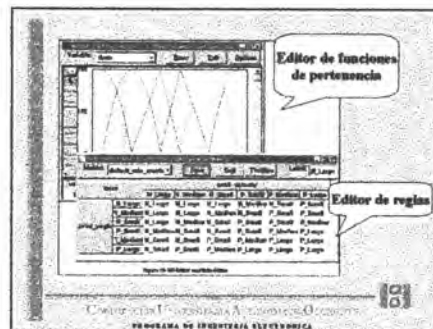
- Editor de matriz de reglas
- Simula microcontroladores (MCU).
- Herramientas de depuración:
 - Trazador
 - Analizador
 - Generador de señales (seno, coseno, exponencial, escalón, rampa y ruido.)
 - Simulador (gráfica entradas y salidas)
 - Compositor (simulación gráfica con bloques)

COMUNICACIÓN Y SISTEMAS AUTOMÁTICOS (COSA) - U. de Sevilla
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 250



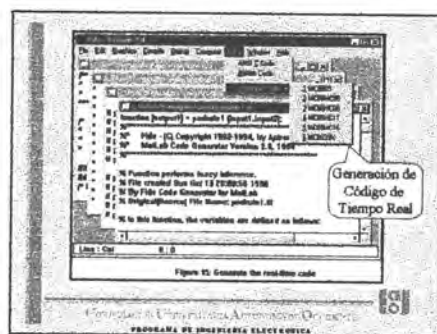
Diapositiva 251



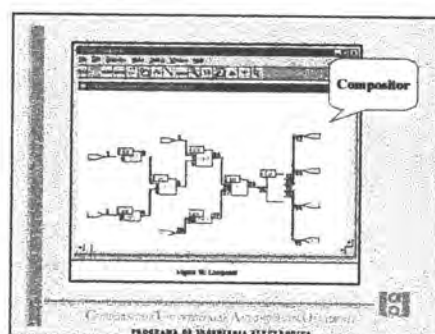
Diapositiva 252



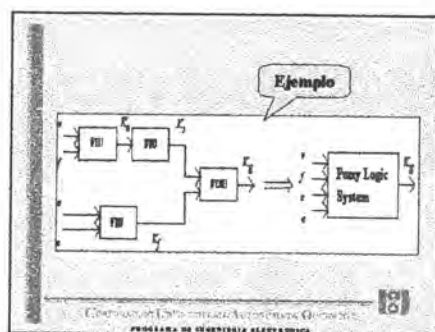
Diapositiva 253



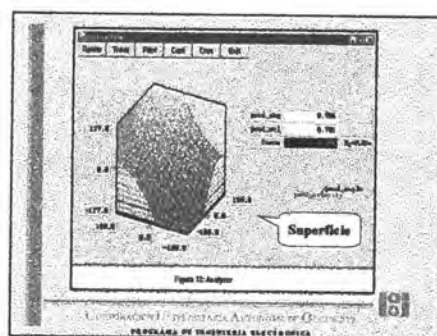
Diapositiva 254



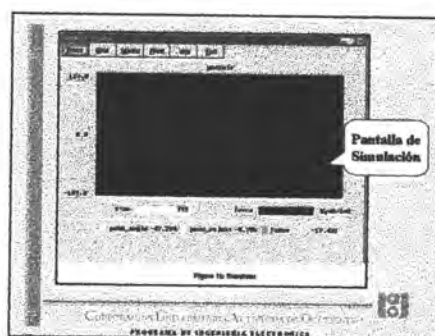
Diapositiva 255



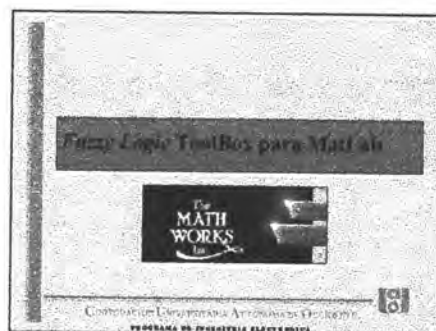
Diapositiva 256



Diapositiva 257



Diapositiva 258



Diapositiva 259

El Toolbox es una colección de funciones construidas en el entorno de cálculo numérico de Matlab. Es un producto de Math Works Inc. que trabaja bajo el ambiente Windows. Esta construido para permitir la creación y edición de sistemas de información difusa con la estructura de Matlab, o si se prefiere se puede integrar el sistema difuso dentro del ambiente gráfico (bloques) de simulación de Simulink. También permite la construcción de programas en C que pueden llamar un sistema difuso construido con Matlab.

Toolbox cuenta con una interfaz gráfica de usuario (GUI) que sirve de herramienta para en la construcción de los sistemas, se puede de igual manera usar la línea de comandos de Matlab si se prefiere.

El Toolbox tiene tres categorías de herramientas:

- Funciones de la línea de comandos
- Herramientas interactivas gráficas (GUI)
- Bloques en Simulink y ejemplos

Universidad Libre y Pontificia Bolivariana
PROGRAMA DE INGENIERIA ELECTRONICA

Diapositiva 260

Distribuidores: Math Works Inc.

Version: 2.0

Interfaces: GUIs

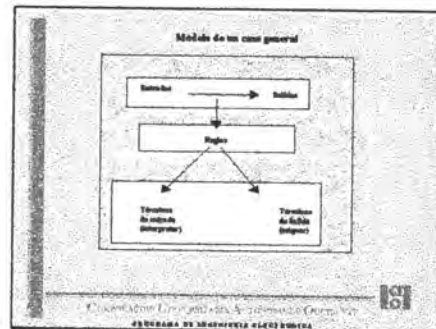
Código Generado: ANSI C, archivos FIS, Simulink

Clientes: Universidades

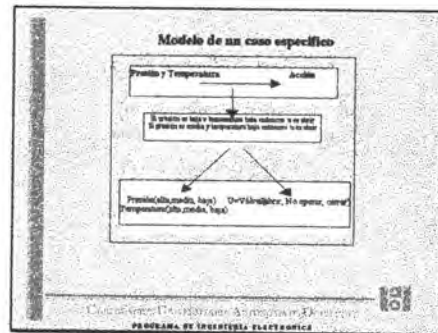
URL: <http://www.mathworks.com>

Universidad Libre y Pontificia Bolivariana
PROGRAMA DE INGENIERIA ELECTRONICA

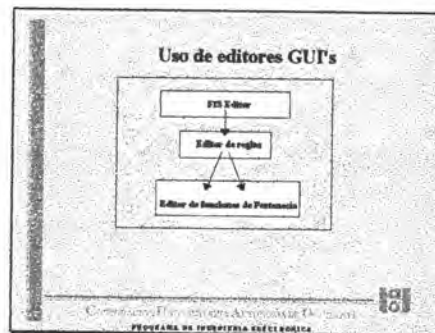
Diapositiva 261



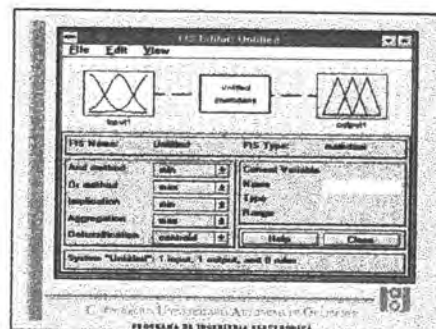
Diapositiva 262



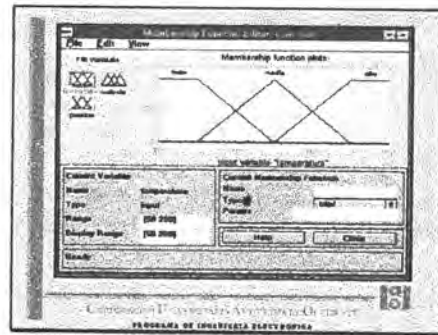
Diapositiva 263



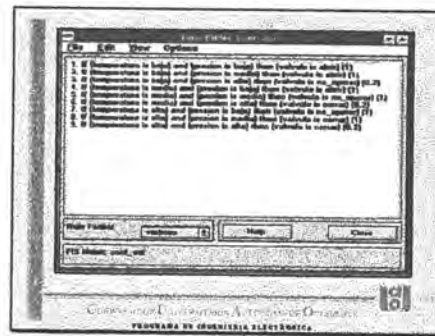
Diapositiva 264



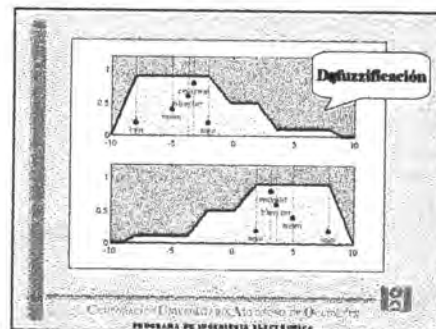
Diapositiva 265



Diapositiva 266



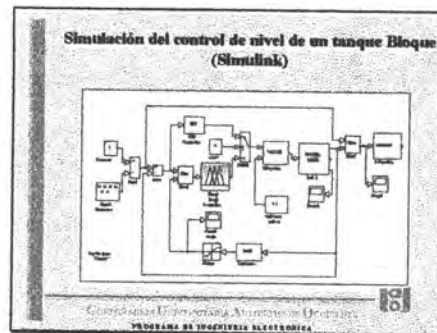
Diapositiva 267



Diapositiva 268



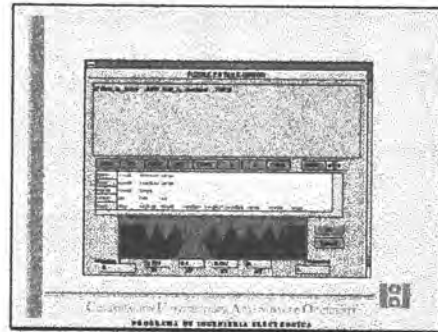
Diapositiva 269



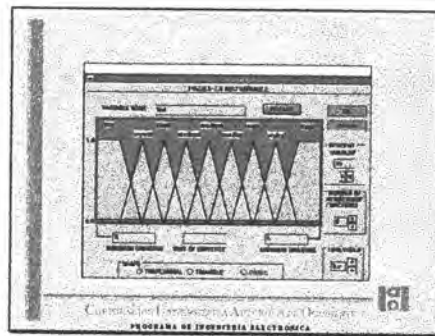
Diapositiva 270



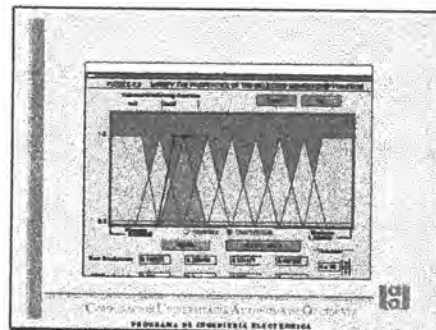
Diapositiva 274



Diapositiva 275



Diapositiva 276



Diapositiva 277

| Clasificación de Software | |
|--|--|
| Clase de software | Características |
| 1. Software de sistema - controla el funcionamiento de la computadora y proporciona la interfaz entre el hardware y el usuario. | <ul style="list-style-type: none"> • controla el flujo de los datos y el flujo de control en la computadora. • controla el flujo de los datos y el flujo de control en la computadora. |
| 2. Software de aplicación - realiza tareas específicas para el usuario. | <ul style="list-style-type: none"> • realizan tareas específicas para el usuario. • realizan tareas específicas para el usuario. |
| 3. Software de programación - ayuda al programador a escribir y ejecutar programas. | <ul style="list-style-type: none"> • ayudan al programador a escribir y ejecutar programas. • ayudan al programador a escribir y ejecutar programas. |

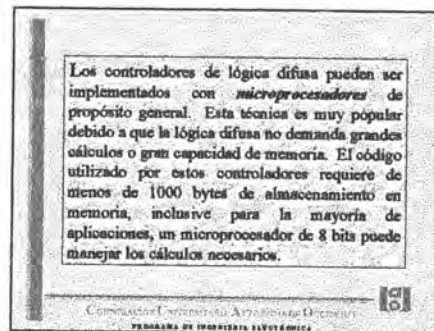
Elaboración: Ing. Juan Carlos A. Rodríguez, D. R. 2010

PROGRAMA DE INGENIERÍA EN SISTEMAS DE COMPUTACIÓN

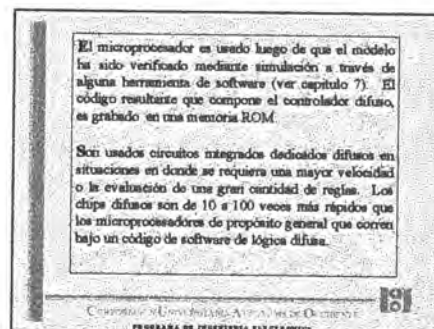
Diapositiva 278



Diapositiva 279



Diapositiva 280



Diapositiva 281

Fuzzy Logic Embedded Controller

Es un completo sistema de desarrollo ofrecido por Hitachi que permite implementar sistemas difusos expertos con los microcontroladores H8/300, H8/300L, H8/300H, y SH. Este paquete incluye dos productos de software difuso el TILShell™ 3.0 (ver capítulo 7) y el MicroFPL.




Corporación Universitaria Autónoma de Occidente
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 282

Características:


- ✓ Su eficiente código hace que se requiera menos memoria para implementar el sistema y que sea transportable a diferentes programas.
- ✓ Permite crear múltiples métodos de defuzzificación, y simples y complejas funciones de pertenencia para el compilador Fuzzy-C.
- ✓ Reduce en 2/3 el tiempo de desarrollo comparado con otros métodos convencionales.
- ✓ Es muy fácil de usar y aprender y no requiere experiencia previa con lógica difusa para su uso.



Corporación Universitaria Autónoma de Occidente
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 283

Togai Infralogic Inc. fabrica toda una línea de controladores difusos dedicados que, según la compañía son 10 veces más rápidos que un 486 de Intel configurado como un controlador difuso. Su RISC (Reduced Instruction-Set Computer) **FC110 IC** maneja hasta 800 reglas y las procesa a una velocidad de 200.000 reglas por segundo.



Corporación Universitaria Autónoma de Occidente
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 284

El controlador difuso de Siemens (81C99) es un coprocesador que sirve de interface para la mayoría de los microprocesadores. Actúa como un acelerador para los microprocesadores de propósito general. Es capaz de procesar las reglas a una velocidad de 7.9 millones de reglas por segundo.

Centro de Investigaciones Avanzadas en Computación
PROGRAMA DE INGENIERÍA ELECTRÓNICA

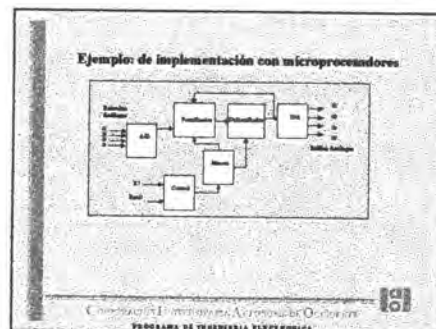
Diapositiva 285

Adaptive Logic (anteriormente American Neurology Inc.)

Fabrica el controlador difuso *NLX 220*. Maneja un máximo de 4 entradas y salidas analógicas. Realiza la conversión de entradas/salidas en tiempo real a través de un chip con convertidores A/D y D/A. Soporta hasta 111 reglas y las procesa a una velocidad de 800000 reglas por segundo. Viene acompañado además, con un sistema de desarrollo, permitiendo a los usuarios simular y configurar el NLX 220 rápidamente para producir nuevos productos.

Centro de Investigaciones Avanzadas en Computación
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 286




Diapositiva 287

Control difuso basado en el 8XC196

Es usado por su bajo costo, pequeño tamaño, manejo de potencia, y altas velocidades de reloj.

Aplicaciones típicas: Control en lazo cerrado y procesamiento de señales como control de motores, discos duros, ABS e instrumentación médica.




Universidad Tecnológica de México
PROGRAMA DE INGENIERÍA ELECTRÓNICA

Diapositiva 288

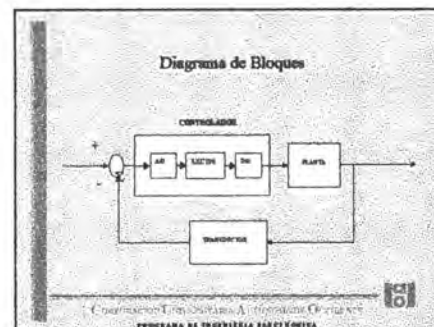
CARACTERÍSTICAS

- El poder del conjunto de instrucciones se basa en que utiliza una arquitectura registro-registro con varios modos de direccionamiento y un rico conjunto de periféricos. La adquisición de datos y el procesamiento son realizados fácil y eficientemente.
- El software puede ser implementado con un mínimo de memoria y rápida ejecución (por encima de 50MHz) localizando el código y los datos en la memoria interna del chip.
- Es viable la utilización del 8XC196 debido a que se reduce el tiempo dedicado para la creación del código.

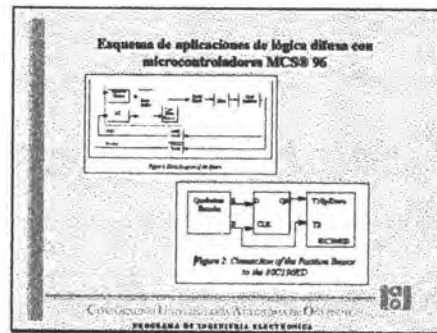


Universidad Tecnológica de México
PROGRAMA DE INGENIERÍA ELECTRÓNICA

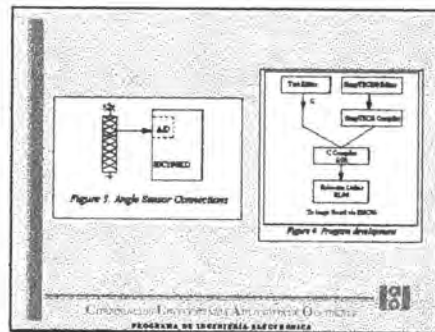
Diapositiva 289



Diapositiva 290



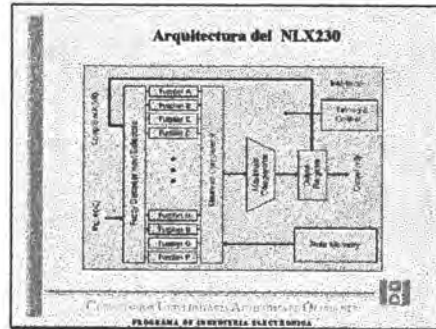
Diapositiva 291



Diapositiva 292



Diapositiva 296



Diapositiva 297



Diapositiva 298



Diapositiva 299

FUZZY-88T-001
PLC Reconfigurable



Descripción:

- Tiene dos entradas y una salida.
- El motor de inferencia implementa relaciones tipo Mamdani's min-max.
- El bloque de defuzzificación usa el método del Centro de Gravedad.
- Cada entrada o salida cuenta de 3 métodos implementados: MZ, MZ, Z, PZ, PZ.
- Implementa 13 reglas fuzzy.
- Los planes de control software el reprogramador del conjunto de reglas.
- Requiere una fuente de voltaje dual de +5V y -5V.
- Se fabrica en tecnología CMOS de 1.8 micrones.
- Tiene una alta velocidad por encima de 6 MPLIPS.
- Empaquetado de 48-Pin tipo DIL.

Centro para Unidades de Automática y Dinámica
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 300

FUZZY-88T-002
Co-processor
Dilene



Descripción:

- Diseñado para aplicaciones en tiempo real y sistemas de alta velocidad.
- Es un co-processor fuzzy diseñado con tecnología VLSI.
- Diseñado con técnicas analógicas y digitales, la parte digital se usa para la configuración del chip, mientras que la parte analógica para los cálculos difusos.
- El motor de inferencia implementa relaciones tipo Mamdani's min-max.
- El bloque de defuzzificación usa el método del Centro de Gravedad.
- Se fabrica en tecnología CMOS de 0.7 micrones.

Centro para Unidades de Automática y Dinámica
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 301


Utiliza un herramienta de software para su configuración a través de un PC

- Tiene una alta velocidad por encima de 4.3 MPLIPS.
- Soporta configuración en línea (on-line).
- Permite funciones de pertenencia tipo S, Z, triangular y trapezoidal.
- Implementa más de 125 reglas (3 antecedentes, 1 consecuente).
- 8-bit de direcciones y 8-bit de bus de datos.
- Empaquetado de 28-Pin tipo DIL.

Centro para Unidades de Automática y Dinámica
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 302

Robot MORIA



Aplicación del coprocesador Fuzzy-set-002

Aplicaciones:

- Transporte dentro de edificios
- Transporte en hospitales
- Tareas de limpieza

Escuela de Ingeniería Electrónica y Electrónica
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 303

Adaptive Logic - AL220 - Stand-Alone Fuzzy Logic Controller



Aplicaciones:

- Control de movimiento y posición
- Control de temperatura
- Manejo de potencia
- Sensores inteligentes
- Control Tipo PID
- Control de motor
- PLC Analógico

Características:

- 4 entradas y salidas analógicas
- Chip software programable (PALCMOS)
- Conversores A/D y D/A internos
- Elimina el uso de los típicos módulos con función de procesamiento externo

Escuela de Ingeniería Electrónica y Electrónica
PROGRAMA DE INGENIERÍA ELÉCTRICA

Diapositiva 304

Fuzzy Logic Trainer (CE124)



Hardware para educación

Parámetros representados son:

- Función de pertenencia
- Inferencia de inferencia
- Operadores lógicos AND, OR, NOT

Características:

- Códice de trabajo en lenguaje (Fuzzy Working Method Code)
- Circuito (Fuzzy Circuit Diagram) (Componentes de Interfaz Fuzzy Rule Implementation)
- Circuito (Fuzzy Circuit Diagram) (Componentes de Fuzzy Rule Table Implementation)

Escuela de Ingeniería Electrónica y Electrónica
PROGRAMA DE INGENIERÍA ELÉCTRICA

